Using Datalog on the Semantic Web

Boris Motik

University of Oxford

March 17, 2010

Table of Contents

1 Introduction

2 Using Datalog for Increased Expressive Power

3 Using Datalog for Efficient Reasoning

4 Conclusion

Semantic Web and DLs

- Semantic Web vision: annotate online resources in a formal language to make data machine-processable
- *Ontologies*: common terminologies for resource description
- Web Ontology Language (OWL)
  - standardized by the W3C
  - OWL 2 is based on the DL $\mathcal{SROIQ}$
  - has a precisely defined first-order semantics
  - enables reasoning over the annotated resources
  - computational properties of reasoning are well known
- Popular OWL reasoners:

FaCT++	[Univ. of Manchester]
Pellet	[Clark&Parsia]
HermiT, CB, REQUIEM	[Univ. of Oxford]
RACER	[Univ. of Hamburg / Concordia Univ.]
CEL	[Univ. of Dresden]
QuONTO	[Univ. of Rome / Univ. of Bolzano]

Description Logics

- *Concepts*: unary predicates
- *Roles*: binary predicates
- *Individuals*: constants

Basic DL $\mathcal{ALC}$: Syntax and Semantics

Interpretation of Roles and Concepts
$(\neg C)^I = x \notin C^I$
$(C \sqcap D)^I = x \in C^I \wedge x \in D^I$
$(C \sqcup D)^I = x \in C^I \vee x \in D^I$
$(\exists R.C)^I = \exists y \in \triangle^I : \langle x, y \rangle \in R^I \wedge y \in C^I$
$(\forall R.C)^I = \forall y \in \triangle^I : \langle x, y \rangle \in R^I \rightarrow y \in C^I$

Interpretation of Axioms and Assertions		
$I \models C \sqsubseteq D$	iff	$C^I \subseteq D^I$
$I \models C(a)$	iff	$a^I \in C^I$
$I \models R(a, b)$	iff	$\langle a^I, b^I \rangle \in R^I$

Reasoning problems:

- KB satisfiability
- concept satisfiability
- concept subsumption
- conjunctive query entailment

DLs vs. Datalog

DLs

- Allow for incomplete information:
  - — existential quantifies (= open domain)
  - — disjunction (= reasoning by case)
- Cannot axiomatize arbitrary non-tree-like relationships
  - — e.g., people living at the same address as their siblings
- Employ the open-world assumption

Datalog

- Usually used in a closed-domain setting
  - — no existential quantifiers, limited to explicitly named objects
- Can axiomatize arbitrary relationships
- Can be extended with closed-world features

Increasing Expressivity

Idea: combine DLs and datalog to increase expressive power

- First-order combinations
  - Adding Datalog rules to DLs
  - Adding DL features to Datalog
- Nonmonotonic combinations
  - Loosely coupled
  - Tightly coupled

Adding Datalog Rules to DLs

Concepts and roles in rules are unary and binary predicates, respectively

The combination overcomes the restriction of the DLs to tree-like axioms

Decidability of Query Answering in DLs

- Existential quantifiers can make the domain infinite
- DLs exhibit a *tree-model property*:
  Each satisfiable KB is satisfiable in a tree-like model.
- Tree-model property can be used to ensure termination

Decidability of Query Answering in Datalog

- Restricted to the explicitly named individuals $\Rightarrow$ finite set!
- No need to restrict the shape of the model to ensure termination

The combination easily gets undecidable

Achieving Decidability

1. Disallow recursion in Datalog rules
   - CARIN [LR98]
2. *Role safety*: restrict at least one variable in each role atom to the explicitly named individuals
   - CARIN [LR98]
3. *DL-safety*: restrict the applicability of the rules to the explicitly named individuals
   - $\mathcal{AL}$-log [DLNS98]
   - DL-safe rules [MSS05]
4. *Tree-like rules*: allow only for rules that do not destroy the tree-model property
   - Description Logic Rules [KRH08]

ADDING DL FEATURES TO DATALOG

Datalog$^{\pm}$ [CGL09]: extension of Datalog with existential quantifiers and equality

- captures database dependencies

Trivially undecidable in unrestricted form

Two decidable restrictions:

- linear rules
  - captures DL-lite and OWL 2 QL
  - query answering is first-order reducible
- weakly guarded rules
  - captures and extends the guarded fragment

Nonmonotonic Combinations

DLs are fragments of first-order logic $\Rightarrow$ cannot axiomatize:

- integrity constraints
  "For every person, the KB must contain a social security number."
- exceptions
  "Without any contrary evidence, the heart is located on the left side of the body."
- closed-world reasoning in general
  "If the flight schedule does not list a flight between BCN and LHR, then no such flight exists."

Main Challenges

First-order + nonmonotonic semantics = ?

The semantics of the integrated KBs should be

- *faithful*
  - if one component is absent, the semantics of the other component should not change
- *useful*
- *decidable*

⇒ Integration is technically very challenging!

- particular problem: dealing with existentially implied objects

LOOSE INTEGRATION

dl-programs [EIL+08]: Datalog rules with queries over a DL KB

$$p(x) \leftarrow DL[R \uplus s;\ C](x)$$

$\Rightarrow$ "temporarily" add the extension of the Datalog predicate *s* to the DL role *R* and query the DL concept *C*

Integration is loose:

- rules do not directly affect the knowledge base
- they are layered as queries over the DL knowledge base

Tight Integration

$\Rightarrow$ Integrated semantics — each component to contributes consequences to the other

- R-hybrid KBs [Ros05], $\mathcal{DL}$+log [Ros06]
  - interpret DL predicates under OWA and Datalog predicates under CWA
- Disjunctive dl-programs [Luk07]
  - uses the DL KB to filter out incompatible models
  - difficult to extend to ASP
- Integration in autoepistemic logic [dBEPT07]
- Integration in MKNF [MR10]

Integration in MKNF

Minimal Knowledge and Negation-as-Failure (MKNF) [Lif94]: a framework integrating many nonmonotonic formalisms

Can be used to integrate DLs with Datalog:

- Captures many related formalisms
  - $\mathcal{DL}$+log, (disjunctive) dl-programs, FO combinations
- Tight: each component contributes consequences to the other component
- Employs DL-safety to achieve decidability
- Proof theory similar to ASP
  $\Rightarrow$ DL KB acts as a filter on the ASP models
- Rules do not “bind” to existentially implied objects

Datalog and Efficiency of Reasoning

DL reasoning is tableau-based:

- difficult to apply techniques such as join order optimizations
- refutation procedure $\Rightarrow$ not suitable for query answering

Common idea: apply database techniques to DL reasoning

DLs allow for recursion $\Rightarrow$ reduce DL KBs to Datalog

Enables the usage of:

- join optimizations
- set-oriented computation of answers
- seminaïve bottom-up evaluation
- magic sets

Resolution-Based Decision Procedures

Reductions are based on resolution decision procedures:

1. Translate a DL KB $\mathcal{K}$ into a set of clauses $\Xi(\mathcal{K})$
2. Saturate $\Xi(\mathcal{K})$ by a resolution calculus that ensures that only finitely many clauses are derived

Types of Clauses Obtained from $\mathcal{ALC}$ KBs

1	$\mathbf{P}(x) \vee R(x, f(x))$
2	$\mathbf{P_1}(x) \vee \mathbf{P_2}(\mathbf{f}(x))$
3	$\mathbf{P_1}(x) \vee \neg R(x, y) \vee \mathbf{P_2}(y)$
4	$\mathbf{P}(\mathbf{a})$
5	$(\neg)R(a, b)$

Legend:

- $\mathbf{P}(x) = (\neg)P_1(x) \vee \ldots \vee (\neg)P_n(x)$
- $\mathbf{P}(\mathbf{f}(x)) = (\neg)P_1(f_1(x)) \vee \ldots \vee (\neg)P_n(f_n(x))$

RESOLUTION-BASED DECISION PROCEDURES

SATURATING $\mathcal{ALC}$ CLAUSES

$$\frac{\mathbf{P_1}(x) \vee \mathbf{P_2}(\mathbf{f}(x)) \vee \neg A(g(x)) \qquad A(x) \vee \mathbf{P_3}(x)}{\mathbf{P_1}(x) \vee \mathbf{P_2}(\mathbf{f}(x)) \vee \mathbf{P_3}(g(x))} \quad \textbf{(2+2=2)}$$

$$\frac{\mathbf{P_1}(x) \vee \neg A(x) \qquad A(x) \vee \mathbf{P_2}(x)}{\mathbf{P_1}(x) \vee \mathbf{P_2}(x)} \quad \textbf{(2+2=2)}$$

$$\frac{\mathbf{P_1}(x) \vee \mathbf{P_2}(\mathbf{f}(x)) \vee \neg A(g(x)) \qquad A(g(x)) \vee \mathbf{P_3}(\mathbf{h}(x)) \vee \mathbf{P_4}(x)}{\mathbf{P_1}(x) \vee \mathbf{P_2}(\mathbf{f}(x)) \vee \mathbf{P_3}(\mathbf{h}(x)) \vee \mathbf{P_4}(x)} \quad \textbf{(2+2=2)}$$

$$\frac{\mathbf{P}(x) \vee R(x, f(x)) \qquad \mathbf{P_1}(x) \vee \neg R(x, y) \vee \mathbf{P_2}(y)}{\mathbf{P}(x) \vee \mathbf{P_1}(x) \vee \mathbf{P_2}(f(x))} \quad \textbf{(1+3=2)}$$

$$\frac{\mathbf{P_1}(\mathbf{a}) \vee \neg A(b) \qquad A(x) \vee \mathbf{P_2}(x)}{\mathbf{P_1}(\mathbf{a}) \vee \mathbf{P_2}(b)} \quad \textbf{(4+2=4)} \qquad \frac{\mathbf{P_1}(\mathbf{a}) \vee \neg A(b) \qquad A(b) \vee \mathbf{P_2}(\mathbf{c})}{\mathbf{P_1}(\mathbf{a}) \vee \mathbf{P_2}(\mathbf{c})} \quad \textbf{(4+4=4)}$$

$$\frac{R(a, b) \qquad \mathbf{P_1}(x) \vee \neg R(x, y) \vee \mathbf{P_2}(y)}{\mathbf{P_1}(a) \vee \mathbf{P_2}(b)} \quad \textbf{(5+3=4)} \qquad \frac{R(a, b) \qquad \neg R(a, b)}{\square} \quad \textbf{(5+5=2)}$$

The set of clauses is *closed under inferences* and *finite*

$\Rightarrow$ Saturation terminates

Reasoning in $\mathcal{EL}$

$\mathcal{EL}$: an expressive DL with a polynomial reasoning procedure

Reducing an $\mathcal{EL}$ KB $\mathcal{K}$ to Datalog [Kaz06]:

1. Translate $\mathcal{K}$ into facts
   - facts describe clauses encountered during resolution
2. Use a Datalog program to simulate resolution inferences
   - the program is fixed—that is, it does not depend on $\mathcal{K}$

Good performance on complex ontologies such as SNOMED

Reducing DLs to (Disjunctive) Datalog

Reduction algorithms for expressive DLs:

- $\mathcal{SHIQ} \Rightarrow$ disjunctive datalog [HMS07]
- $\mathcal{ELHIO}$ + conjunctive query $\Rightarrow$ datalog [PUMH09]

Reduction overview:

1. Translate the TBox of $\mathcal{K}$ into clauses $\Xi(\mathcal{K})$
2. Saturate $\Xi(\mathcal{K})$ (+ query) using a suitable resolution variant
   - derives all “relevant” consequences
3. Eliminate clauses containing function symbols
   - possible because we have derived “relevant” consequences
4. Convert resulting clauses into a Datalog program
5. Append ABox

Reduction Example

Description Logic Knowledge Base		
$A \sqsubseteq \exists R.B$	$B \sqsubseteq C$	$\exists R.C \sqsubseteq D$

Translation into Clauses		
$\neg A(x) \vee R(x, f(x))$	$\neg A(x) \vee B(f(x))$	$\neg B(x) \vee C(x)$
	$D(x) \vee \neg R(x, y) \vee \neg C(y)$	

Saturation		
$\neg A(x) \vee C(f(x))$	$D(x) \vee \neg A(x) \vee \neg C(f(x))$	$D(x) \vee \neg A(x)$

Reduction Result		
$B(x) \rightarrow C(x)$	$R(x, y) \wedge C(y) \rightarrow D(x)$	$A(x) \rightarrow D(x)$

- Rule $A(x) \rightarrow D(x)$ acts as a “shortcut” for inferences involving existentially implied objects

Reduction via OBBDs

Alternative reduction of $\mathcal{SHIQ}$ to disjunctive datalog [RKH08]

1. Translate the TBox into an OBBD
2. Translate the OBBD into a disjunctive datalog program
3. Append the ABox

No comprehensive evaluation so far

Research Directions

- Tighten integration under a nonmonotonic semantics
  - dealing with existentials under CWA
- Develop DLs that correspond to efficient Datalog fragments
  - linear Datalog, monadic Datalog, ...
- Exploit structural properties of KBs to optimize translation
  - treewidth, ...
- Develop efficient implementations

References

[CGL09] A. Calì, G. Gottlob, and T. Lukasiewicz.
A general datalog-based framework for tractable query answering over ontologies.
In *Proc. PODS*, pages 77–86, 2009.

[dBEPT07] J. de Bruijn, T. Eiter, A. Polleres, and H. Tompits.
Embedding Non-Ground Logic Programs into Autoepistemic Logic for Knowledge-Base Combination.
In *Proc. IJCAI*, pages 304–309, 2007.

[DLNS98] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf.
AL-log: Integrating Datalog and Description Logics.
Journal of Intelligent Information Systems, 10(3):227–252, 1998.

[EIL+08] T. Eiter, G. Ianni, T. Lukasiewicz, R. Schindlauer, and H. Tompits.
Combining answer set programming with description logics for the Semantic Web.
AIJ, 172(12–13):1495–1539, 2008.

[HMS07] U. Hustadt, B. Motik, and U. Sattler.
Reasoning in Description Logics by a Reduction to Disjunctive Datalog.
JAR, 39(3):351–384, 2007.

[Kaz06] Y. Kazakov.
Saturation-Based Decision Procedures for Extensions of the Guarded Fragment.
PhD thesis, Univesität des Saarlandes, Germany, 2006.

[KRH08] M. Krötzsch, S. Rudolph, and P. Hitzler.
ELP: Tractable Rules for OWL 2.
In *Proc. ISWC*, pages 649–664, 2008.

[Lif94] V. Lifschitz.
Minimal Belief and Negation as Failure.
AIJ, 70(1–2):53–72, 1994.

REFERENCES

[LR98] A. Y. Levy and M.-C. Rousset.
Combining Horn Rules and Description Logics in CARIN.
AIJ, 104(1–2):165–209, 1998.

[Luk07] T. Lukasiewicz.
A Novel Combination of Answer Set Programming with Description Logics for the Semantic Web.
In *Proc. ESWC*, pages 384–398, 2007.

[MR10] Boris Motik and Riccardo Rosati.
Reconciling Description Logics and Rules.
J. ACM, 2010.
Accepted for publication.

[MSS05] Boris Motik, Ulrike Sattler, and Rudi Studer.
Query Answering for OWL-DL with Rules.
JWS, 3(1):41–60, 2005.

[PUMH09] H. Pérez-Urbina, B. Motik, and I. Horrocks.
Tractable Query Answering and Rewriting under Description Logic Constraints.
JAL, 8(2):151–232, 2009.

[RKH08] S. Rudolph, M. Krötzsch, and P. Hitzler.
Description logic reasoning with decision diagrams: Compiling shiq to disjunctive datalog.
In *Proc. ISWC*, pages 435–450, 2008.

[Ros05] R. Rosati.
On the decidability and complexity of integrating ontologies and rules.
JWS, 3(1):61–73, 2005.

[Ros06] R. Rosati.
$\mathcal{DL}+log$: A Tight Integration of Description Logics and Disjunctive Datalog.
In *Proc. KR*, pages 68–78, 2006.