Extending Logic Programs with Description Logic Expressions for the Semantic Web

Yi-Dong Shen

Chinese Academy of Sciences
Beijing, China

http://lcs.ios.ac.cn/~ydshen

Kewen Wang

Griffith University
Brisbane, Australia

http://www.ict.griffith.edu.au/~kewen

ISWC 2011, Bonn, Germany

Outline

I. Background and Motivation

II. Logic Programs and DL Knowledge Bases

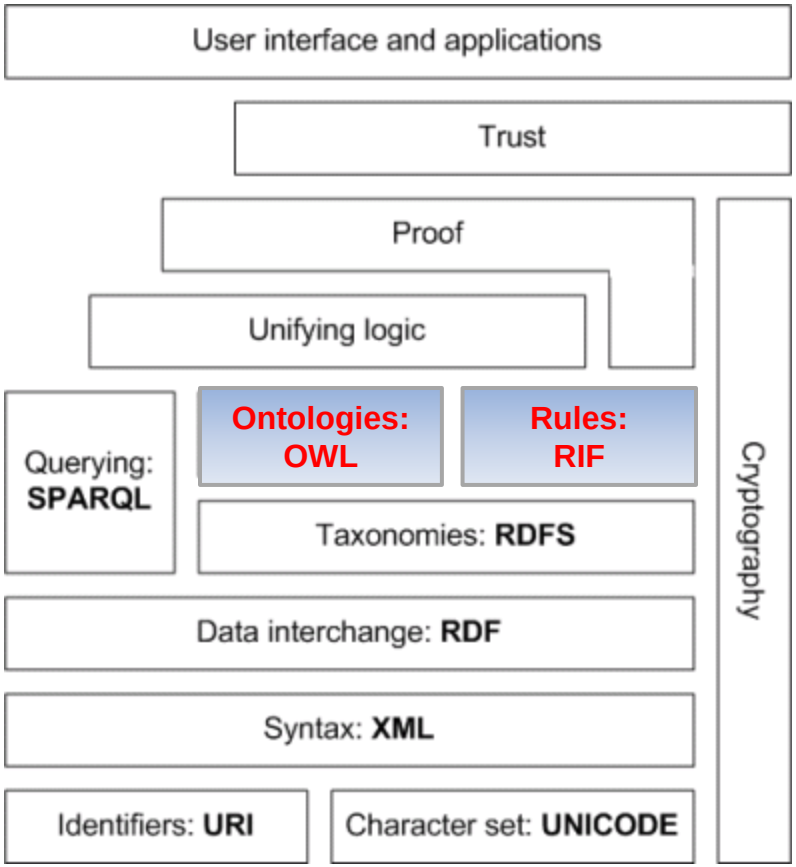
III. Normal DL Logic Programs

- Syntax
- Semantics

IV. Related Work

V. Summary and Future Work

Semantic Web Stack



Ontologies and Rules

- **Ontologies** describe terminological knowledge.
- **Rules** model constraints and exceptions over the ontologies.
- The two components provide complementary descriptions of the same problem domain, so it is necessary to integrate them in some ways (a **unifying logic**).

Logic Program Rules

- The integration depends on what knowledge representation formalisms are used to represent rules.
- Logic programming is a KR language paradigm widely used for representing and reasoning with rules.
- Therefore, recently much attention has been directed to using logic programs to represent rules in the integration for the Semantic Web.

Logic Programs with DL Expressions

- A normal logic program consists of **if-then rules**

$$H \leftarrow A_1, \cdots, A_m, not\ B_1, \cdots, not\ B_n$$

where H, each A_i and B_i are atoms.

- Under the integration, logic programs are extended by **allowing description logic (DL) expressions to appear in rules**, so that logic programs have access to DL knowledge bases (ontologies) and thus are able to reason with ontologies in the Semantic Web.

Major Existing Proposals

1. Description logic programs (or dl-programs)

   (Eiter et al., AIJ2008).

2. $\mathcal{DL}$+log

   (Rosati, KR2006).

3. Disjunctive dl-programs

   (Lukasiewicz, TKDE2010).

dl-Programs (Eiter et al., AIJ2008)

- Given an external DL knowledge base L, a dl-program extends a normal logic program Π by
  - adding dl-atoms to rule bodies as an interface to access to L.
- L and Π share no predicate symbols in their vocabularies, so a mapping of predicate symbols between L and Π is required.
- DL atoms are not allowed to appear in rule heads.

$\mathcal{DL}$+log (Rosati, KR2006)

- $\mathcal{DL}$+log extends a normal logic program Π by
  - letting L and Π share some predicate symbols in their vocabularies, and
  - allowing atomic DL expressions (i.e. atomic concepts and atomic roles) to appear either in bodies or heads of rules without using any predicate mapping operators.
- One restriction is that DL expressions are not allowed to appear behind the negation operator not.

Disjunctive dl-Programs
(Lukasiewicz, TKDE2010)

- A disjunctive dl-program extends a normal logic program Π by
  - letting L and Π share some predicate symbols in their vocabularies, and
  - allowing atomic DL expressions to appear anywhere in a rule.

Complementary Features

- In syntax
  - dl-programs allow arbitrary DL expressions in rule bodies.
  - $\mathcal{DL}$+log and disjunctive dl-programs allow atomic DL expressions in rule heads.
- In semantics
  - dl-programs and $\mathcal{DL}$+log: DL concepts and roles occurring in Π are all interpreted against L under the first-order semantics.
  - disjunctive dl-programs: DL concepts and roles occurring in Π are all included in the Herbrand base of Π and interpreted under the answer set semantics.

Complementary Features

- It is desirable to have a new extension of logic programs with DL expressions, which combines the complementary features of dl-programs, $\mathcal{DL}$+log and disjunctive dl-programs.
- **This motivates the work of our current paper.**

Our Contributions

- We propose a new extension, called a normal DL logic program, which combines the complementary features of dl-programs, $\mathcal{DL}$+log and disjunctive dl-programs by
  - allowing arbitrary DL expressions to appear in rule bodies and atomic DL expressions in rule heads;
  - allowing to interpret DL concepts and roles occurring in Π flexibly either in first-order semantics or answer set semantics; and
  - having a well-supported answer set semantics, so that its answer sets are free of circular justifications.

Outline

I. Background and Motivation

II. Logic Programs and DL Knowledge Bases

III. Normal DL Logic Programs

- Syntax
- Semantics

IV. Related Work

V. Summary and Future Work

Logic Programs

- A vocabulary $\Sigma_\Pi = (\mathbf{P}, \mathbf{C})$
  - $\mathbf{P}$: a finite set of predicate symbols.
  - $\mathbf{C}$: a nonempty finite set of constants.
- A normal logic program Π is a finite set of rules

$$H \leftarrow A_1, \cdots, A_m, not\ B_1, \cdots, not\ B_n$$

where H, A_i and B_i are atoms built over Σ_Π.

- $\mathrm{ground}(\Pi)$: all ground instances of Π obtained by replacing all variables in Π with constants in $\mathbf{C}$.

Logic Programs

- Herbrand base HB_Π

  All ground atoms $\mathrm{p}(t_1, \cdots, t_m)$, where $\mathrm{p} \in \mathbf{P}$ occurs in Π and $t_i \in \mathbf{C}$.

- Herbrand interpretation I

  Any subset of HB_Π. Let $I^- = HB_\Pi \backslash I$ and $\neg I^- = \{\neg a | a \in I^-\}$.

- Standard answer set semantics (Gelfond and Lifschitz, NJC1991)

  I is an answer set of Π if I is the least model of Π^I, where

  $\Pi^I = \{a \leftarrow pos(r) \mid a \leftarrow pos(r), neg(r) \in \mathrm{ground}(\Pi) \text{ and } I \text{ satisfies } neg(r)\}$.

DL Knowledge Bases

- A description logic $\mathcal{SHOIN}$ (Horrocks et al., JWS2003) with a vocabulary $\Sigma_L = (\mathbf{A} \cup \mathbf{R}, \mathbf{I})$
  - **A, R, I:** atomic concepts, atomic roles, and individuals.
- A DL knowledge base L is a finite set of axioms over Σ_L
  - $C \sqsubseteq D$: concept inclusion axiom.
  - $R \sqsubseteq R_1$: role inclusion axiom.
  - $Trans(R)$: transitivity axiom.
  - $C(a)$: concept membership axiom.
  - $R(a, b)$: role membership axiom.
  - $= (a, b)$ or $\neq (a, b)$:.equality/inequality axiom.

  where C, D are concepts; R, R_1 atomic roles; a, b individuals.

DL Knowledge Bases

- Since DLs are fragments of first-order logic, a DL knowledge base L has first-order semantics.
- L is consistent (or satisfiable) if L has a first-order model.
- For an axiom F, L entails F, denoted $L \models F$, if all first-order models of L are first-order models of F.

Outline

I. Background and Motivation

II. Logic Programs and DL Knowledge Bases

III. Normal DL Logic Programs

- Syntax
- Semantics

IV. Related Work

V. Summary and Future Work

Syntax

- Let L be a DL knowledge base built over $\Sigma_L = (\mathbf{A} \cup \mathbf{R}, \mathbf{I})$
- We develop a logic program Π over $\Sigma_\Pi = (\mathbf{P}, \mathbf{C})$ with DL expressions relative to L, called a normal DL logic program, where
  - $\mathbf{C} \subseteq \mathbf{I}$, i.e. constants come from individuals; and
  - L and Π share a set $\mathbf{\Omega} = \mathbf{P} \cap (\mathbf{A} \cup \mathbf{R})$ of predicate symbols (atomic concepts and roles).

Note: Concepts and roles in $\mathbf{\Omega}$ shared by Π are intended to be interpreted in Herbrand models under answer set semantics.

DL Expressions

- A DL expression or DL query (Eiter et al., AIJ2008), which is allowed to appear in rules of a logic program, is
  - a concept inclusion axiom $C \sqsubseteq D$ or its negation;
  - $C(t)$ or $\neg C(t)$, where C is a concept and t a term (variable or constant);
  - $R(t_1, t_2)$ or $\neg R(t_1, t_2)$, where R is an atomic role or its inverse, and t_1, t_2 are terms; or
  - $=(t_1, t_2)$ or $\neq (t_1, t_2)$, where t_1, t_2 are terms.
- An atomic DL expression is either $C(t)$ or $R(t_1, t_2)$, where C is an atomic concept and R is an atomic role.

Normal DL Logic Programs

- **Definition** Given a DL knowledge base L, a normal DL logic program Π with DL expressions relative to L is a finite set of rules

$$H \leftarrow A_1, \cdots, A_m, not\ B_1, \cdots, not\ B_n$$

where H is an atom, and each A_i and B_i are either atoms or DL expressions.

Normal DL Logic Programs

Example 1 Let $L = \{E \sqsubseteq F\}$ and

$$\Pi: \; A(g),$$
$$B(X) \leftarrow C(X),$$
$$C(X) \leftarrow A(X), (\neg C \sqcup B \sqcup F)(X).$$

Let $\mathbf{P} = \{A, B, C\}$, $\mathbf{C} = \{g\}$ and $\mathbf{\Omega} = \{B, C\}$. Then

$$\text{ground}(\Pi): \; A(g),$$
$$B(g) \leftarrow C(g),$$
$$C(g) \leftarrow A(g), (\neg C \sqcup B \sqcup F)(g).$$

Semantics

- Herbrand base HB_{Π} of Π relative to L

  All ground atoms $\mathrm{p}(t_1, \cdots, t_m)$,

  where $\mathrm{p} \in \mathbf{P}$ occurs in Π or L and $t_i \in \mathbf{C}$.

- For a Herbrand interpretation $I \subseteq HB_{\Pi}$, let
  - $I|_{\Omega} = \{ \mathrm{A} \in I \mid \text{the predicate symbol of } A \text{ is in } \Omega \}$.
  - $I^{-}|_{\Omega} = \{ \mathrm{A} \in I^{-} \mid \text{the predicate symbol of } A \text{ is in } \Omega \}$.
- I is consistent with L if $L \cup I|_{\Omega} \cup \neg I^{-}|_{\Omega}$ is consistent.

Extended Satisfaction $I \models_L A$

- Satisfaction of a Herbrand interpretation I relative to L

Definition I satisfies A under L, denoted $I \models_L A$:

- For a ground atom $A \in HB_\Pi$, which is not an atomic DL expression, $I \models_L A$ if $A \in I$.
- For a ground DL expression A, $I \models_L A$ if $L \cup I|_\Omega \cup \neg I^-|_\Omega \models A$.
- For a ground atom or a ground DL expression A, $I \models_L not\ A$ if $I \not\models_L A$.

Herbrand Models

- **Definition** A Herbrand interpretation I is a model of Π relative to L if I is consistent with L and $I \models_L r$ for all rules $r \in \text{ground}(\Pi)$.

up to Satisfaction $(E, I) \models_L A$

- To define well-supported models for normal DL logic programs, we introduce E up to I satisfies A under L, denoted $(E, I) \models_L A$:

Definition Let A be a ground atom or DL expression and $E \subseteq I \subseteq HB_\Pi$.

- $(E, I) \models_L A$ if for every F with $E \subseteq F \subseteq I$, $F \models_L A$;
- $(E, I) \models_L not\ A$ if for no such F, $F \models_L A$.

- $(E, I) \models_L A$ implies that the truth of A depends only on E and I^-, and is independent of $I \backslash \mathrm{E}$.
- For instance, if $E = \{a\}$, $I = \{a, b, c\}$ and $A = a \wedge \neg d$,

then for every F with $E \subseteq F \subseteq I$, $F \models_L A$. Thus, $(E, I) \models_L A$.

Monotonicity of $(E, I) \models_L A$

Theorem Let A be a ground atom or DL expression, and

$E_1 \subseteq E_2 \subseteq I \subseteq HB_\Pi$.

- If $(E_1, I) \models_L A$, then $(E_2, I) \models_L A$;
- If $(E_1, I) \models_L not\ A$, then $(E_2, I) \models_L not\ A$.

Well-Supported Models

- **Definition** A model *I* of a normal DL logic program is well-supported if there is a level mapping on *I* such that for every $a \in I$ there exists $E \subset I$, where the level of each $b \in E$ is below the level of a, such that
  - $L \cup E|_{\Omega} \cup \neg I^{-}|_{\Omega} \models a$ or
  - there is $a \leftarrow body(r) \in \text{ground}(\Pi)$ such that $(E, I) \models_L body(r)$.

A Fixpoint Semantics

- Consequence operator $T_{\Pi}(E, I)$

Definition Let Π be a normal DL logic program relative to a DL knowledge base L, and I a Herbrand interpretation consistent with L. For $E \subseteq I$, define

$$T_{\Pi}(E, I) = \{a | a \leftarrow body(r) \in \text{ground}(\Pi) \text{ and } (E, I) \models_L body(r)\}$$

A Fixpoint Semantics

- Monotonicity property of $T_{\Pi}(E, I)$

Theorem Let I be a model of Π relative to L. For any $E_1 \subseteq E_2 \subseteq I$, $T_{\Pi}(E_1,\ I) \subseteq T_{\Pi}(E_2,\ I) \subseteq I$.

A Fixpoint Semantics

- We use $T_{\Pi}^{\alpha}(\emptyset, I)$ to define a fixpoint semantics

Definition Let I be a model of a normal DL logic program Π relative to a DL knowledge base L. I is an answer set of Π relative to L if for every $a \in I$, either

- $a \in T_{\Pi}^{\alpha}(\emptyset, I)$, or
- $L \cup T_{\Pi}^{\alpha}(\emptyset, I)|_{\Omega} \cup \neg I^{-}|_{\Omega} \models a$.

A Fixpoint Semantics

Example 1 Let $L = \{E \sqsubseteq F\}$ and

Π: $A(g)$,

$B(X) \leftarrow C(X)$,

$C(X) \leftarrow A(X), (\neg C \sqcup B \sqcup F)(X)$.

Let $\mathbf{P} = \{A, B, C\}$, $\mathbf{C} = \{g\}$ and $\boldsymbol{\Omega} = \{B, C\}$. Then

- $HB_{\Pi} = \{A(g), B(g), C(g)\}$.
- $I = \{A(g), B(g), C(g)\}$ is the only model of Π relative to L.
- I is not an answer set of Π relative to L since $T_{\Pi}^{\alpha}(\emptyset, I) = \{A(g)\}$, and neither $B(g) \in T_{\Pi}^{\alpha}(\emptyset, I)$ nor $L \cup T_{\Pi}^{\alpha}(\emptyset, I)|_{\Omega} \cup \neg I^{-}|_{\Omega} \models B(g)$.

Note: I is not a well-supported model of Π relative to L.

Properties of the Semantics

- Answer sets are minimal and well-supported models

Theorem 1 If I is an answer set of Π relative to L, then I is an minimal model of Π relative to L.

Theorem 2 I is an answer set of Π relative to L iff I is a well-supported model of Π relative to L.

Theorem 3 Let $L = \emptyset$ and Π be a normal DL logic program without DL expressions. I is an answer set of Π relative to L iff I is

Decidability of the Semantics

- The decidability of computing answer sets of Π relative to L depends on the decidability of satisfiability of L.
- Since DLs are fragments of first-order logic, the satisfiability of L is undecidable in general cases.
- If L is built from $\mathcal{SHOIN}$ or $\mathcal{SROIQ}$, the satisfiability of L is decidable (Horrocks et al., JWS2003; KR2006). Therefore, it is decidable to compute all such answer sets.

Outline

I. Background and Motivation

II. Logic Programs and DL Knowledge Bases

III. Normal DL Logic Programs

- Syntax
- Semantics

IV. Related Work

V. Summary and Future Work

Related Work

- We focus on enhancing logic program rules with DL expressions so that logic programs are able to access to external DL knowledge bases and thus able to reason with ontologies in the Semantic Web.
- This differs fundamentally from modal logic based embeddings of rules and DLs, such as (de Bruijn et al., KR2008) and (Motik and Rosati, JACM2010), which
  - transform rules Π and DL axioms L into modal logic formulas Π' and L', and
  - use the modal logic semantics of $\Pi' \cup L'$ as the semantics of Π and L.

Related Work

- Three closely related approaches to extending logic programs with DL expressions:
  - dl-programs (Eiter et al., AIJ2008),
  - $\mathcal{DL}$+log (Rosati, KR2006) and its variant called *guarded hybrid knowledge bases* (Heymans et al., TPLP2008).
  - disjunctive dl-programs (Lukasiewicz, TKDE2010).

Related Work

		dl-programs	$\mathcal{DL}$+log	Disjunctive dl-programs	Normal DL logic programs
Arbitrary or atomic DL expressions	Rule body	arbitrary	atomic	atomic	arbitrary
	Rule head	X	atomic	atomic	atomic
Need predicate mapping		yes	no	no	no
Interpretation of DL predicates occurring in rules		first-order only*	first-order only*	Herbrand only**	flexible***
Well-supported semantics free of circular justifications		yes (Shen, IJCAI2011)	no	no	yes

* First-order interpretation against the external DL knowledge base.

** Herbrand interpretation under the answer set semantics.

*** When being included in Ω, Herbrand only; otherwise, first-order only.

Outline

I. Background and Motivation

II. Logic Programs and DL Knowledge Bases

III. Normal DL Logic Programs

- Syntax
- Semantics

IV. Related Work

V. Summary and Future Work

Summary

- We presented normal DL logic programs, a new extension of logic programs with DL expressions, which combine the complementary features of dl-programs, $\mathcal{DL}$+log and disjunctive dl-programs by
  - allowing arbitrary DL expressions to appear in rule bodies and atomic DL expressions in rule heads;
  - allowing to interpret DL concepts and roles occurring in rules either in first-order semantics or answer set semantics; and
  - having a well-supported answer set semantics, so that their answer sets are free of circular justifications.

Future Work

- Study computational properties of the semantics for normal DL logic programs w.r.t. different DLs.
- Extend the work to disjunctive DL logic programs, where rule heads are a disjunction of atoms or atomic DL expressions.
- Methods for implementing the fixpoint semantics for normal DL logic programs present interesting future work.

Thanks !

Yi-Dong Shen
ydshen@ios.ac.cn
http://lcs.ios.ac.cn/~ydshen