Well-Supported Semantics for Description Logic Programs

Yi-Dong Shen
Institute of Software, Chinese Academy of Sciences, Beijing, China
http://lcs.ios.ac.cn/~ydshen

IJCAI 2011, Barcelona, Spain
Outline

I. Background and Motivation
II. DL-Programs
III. Well-Supported Models
IV. Well-Supported Answer Set Semantics
V. Related Work
VI. Summary and Future Work
Semantic Web Stack

- User interface and applications
- Trust
- Proof
- Unifying logic
- Querying: SPARQL
- Ontologies: OWL
- Rules: RIF/SWRL
- Taxonomies: RDFS
- Data interchange: RDF
- Syntax: XML
- Identifiers: URI
- Character set: UNICODE
Integration in the Semantic Web

- **Ontologies** describe terminological knowledge.
- **Rules** model constraints and exceptions over the ontologies.
- They provide complementary descriptions of the same problem domain, so a **unifying logic** is used to
  - integrate the two components, and
  - study the semantic properties of the integrated knowledge base.
Three Forms of Integration

- **Loose integration**
  - Ontologies and rules share no predicate symbols (Eiter et al. 2008, AIJ).

- **Tight (or Hybrid) integration**
  - Ontologies and rules share some predicate symbols (Rosati 2006, KR; Lukasiewicz 2010, TKDE).

- **Full integration**
  - Ontologies and rules share the same vocabulary (de Bruijn et al. 2008, KR; Motik and Rosati 2010, JACM).
We consider a loose integration, called Description logic programs (or DL-programs) (Eiter et al. 2008, AIJ)

A DL-program is $KB = (L, R)$

- $L$: a DL knowledge base (ontologies).
- $R$: an extended logic program under the answer set semantics.
Semantic Issues with DL-Programs

- **Weak answer set semantics** (Eiter et al. 2008, AIJ)
  - The authors noted that an obvious disadvantage of the semantics is that it may produce counterintuitive answer sets with *circular justifications* by self-supporting loops.

- **Strong answer set semantics** (Eiter et al. 2008, AIJ)
  - We observed that the problem of *circular justifications* persists in this semantics.

- **FLP answer set semantics** (Eiter et al. 2005, IJCAI)
  - We observed that the problem of *circular justifications* persists in this semantics.
Semantic Issues with DL-Programs

Therefore, it presents an interesting yet challenging open problem to develop a new semantics for DL-programs, which produces answer sets free of circular justifications.
Circular Justifications

- A model $I$ of a logic program $R$ is **circularly justified** if the truth of some $a \in I$ is supported by itself in $I$.

- **Examples**
  
  1. Consider a logic program $R = \{a \leftarrow b. \ b \leftarrow a\}$ and let $I = \{a, b\}$.  
     
     $a \in I$ is circularly justified by a self-supporting loop: $a \leftarrow b \leftarrow a$

  2. Consider a DL-program $KB = (L, R)$ from (Eiter et al. 2008, AIJ), where $L = \emptyset$ and $R = \{p(a) \leftarrow DL[c \cup p; c](a)\}$. Let $I = \{p(a)\}$.  
     
     $p(a) \in I$ is circularly justified by a self-supporting loop:
     
     $p(a) \leftarrow DL[c \cup p; c](a) \leftarrow p(a)$
Fages’ Well-Supportedness Condition

- For normal logic programs, the problem of circular justifications is elegantly handled by Fages’ well-supportedness condition (Fages 1994, JMLCS).
- It defines a level mapping, which prevents well-supported models from circular justifications.
- It is a key property to characterize the standard answer set semantics (Gelfond and Lifschitz 1991, NJC):
  - A model of a normal logic program is an answer set under the standard answer set semantics iff it is well-supported (Fages 1994, JMLCS).
Fages’ Well-Supportedness Condition

- Can we extend Fages’ well-supportedness condition from normal logic programs to DL-programs to overcome circular justifications?
- Our answer is Yes.
Our Contributions

- We solve the semantic problem of circular justifications with DL-programs by
  - extending Fages’ well-supportedness condition from normal logic programs to DL-programs, and
  - defining a well-supported semantics for DL-programs, which produces answer sets free of circular justifications.
Outline

I. Background and Motivation

II. DL-Programs

III. Well-Supported Models

IV. Well-Supported Answer Set Semantics

V. Related Work

VI. Summary and Future Work
Notation

- **A DL-program** is $KB = (L, R)$

- $L$: a **DL knowledge base** built over $\Sigma_L = (A \cup R, I)$
  - $A$, $R$, $I$: atomic concepts, atomic roles, and individuals.

- $R$: a **rule base** built over $\Sigma_R = (P, C)$
  - $P$, $C$: predicate symbols, and constants
  - $P \cap (A \cup R) = \emptyset$, and $C \subseteq I$
  - $HB_R$: Herbrand base of $R$ built over $\Sigma_R$

- **ground($R$)**: ground instances (relative to $HB_R$) of all rules in $R$
Notation

- $R$ consists of rules of the form
  
  $$H \leftarrow A_1, \cdots, A_m, \text{not } B_1, \cdots, \text{not } B_n$$

  where $H$ is an atom, and each $A_i$ and $B_i$ are atoms or dl-atoms

- A dl-atom is an interface between $L$ and $R$:

  $$DL[S_1 \ op_1 \ p_1, \cdots, S_m \ op_m \ p_m; \ Q](t)$$

  - each $S_i$ is a concept or role built from $A \cup R$, each $p_i \in P$ is a predicate symbol, $Q(t)$ is a dl-query and $op_i \in \{ \cup, \cap \}$
Satisfaction Relation $\models_L$

**Definition** (Eiter et al. 2008, AlJ) Let $KB = (L, R)$ and $I$ be an interpretation. Define *satisfaction under $L$*, denoted $\models_L$, as follows:

1. For a ground atom $a \in HB_R$, $I \models_L a$ if $a \in I$.

2. For a ground dl-atom $A = DL[S_1 op_1 p_1, \ldots, S_m op_m p_m; Q](t)$, 

   $I \models_L A$ if $L \cup \bigcup_{i=1}^{m} A_i \models Q(t)$, where

   $$A_i = \begin{cases} 
   \{S_i(e) \mid p_i(e) \in I\}, & \text{if } op_i = \cup; \\
   \{-S_i(e) \mid p_i(e) \in I\}, & \text{if } op_i = \cup; \\
   \{-S_i(e) \mid p_i(e) \not\in I\}, & \text{if } op_i = \cap.
   \end{cases}$$

*** Any $I \subseteq HB_R$ is an interpretation of $KB = (L, R)$. Let $I^- = HB_R \setminus I$ and $\neg I^- = \{\neg a \mid a \in I^-\}$
**Program Transformation Reducts**

- Given an interpretation $I$, FLP reduct $fR^I_L$ is obtained from $\text{ground}(R)$ by
  deleting every rule $r$ with $I \not\models_L \text{body}(r)$.

- **Weak transformation reduct** $wR^I_L$ is obtained from $fR^I_L$ by
  deleting all negative literals and all dl-atoms.

- **Strong transformation reduct** $sR^I_L$ is obtained from $fR^I_L$ by
  deleting all negative literals and all nonmonotonic dl-atoms.

*** A ground dl-atom $A$ is **monotonic**

  if for any $I \subseteq J \subseteq HB_R$, $I \models_L A$ implies $J \models_L A$. 
Three Semantics of DL-Programs

- Weak/strong/FLP answer set semantics
  A model $I$ of $KB = (L, R)$ is a weak (resp. strong and FLP) answer set if $I$ is a minimal model of $wR^I_L$ (resp. $sR^I_L$ and $fR^I_L$) (Eiter et al. 2008, AIJ; Eiter et al. 2005, IJCAI).

- FLP answer sets are minimal models, but weak/strong answer sets may not.
Circular Justification Problem

- The three answer set semantics suffer from the problem of **circular justifications**.

- **Example**  Consider a DL-program $KB = (L, R)$, where $L = \emptyset$ and

  $$R: \quad p(a) \leftarrow q(a)$$

  $$q(a) \leftarrow DL[c \cup p, b \cap q; c \cap \neg b](a)$$

  $I = \{p(a), q(b)\}$ is the only model of $KB$. It is also a weak, a strong, and an FLP answer set. $p(a) \in I$ is circularly justified by a self-supporting loop:

  $$p(a) \leftarrow q(a) \leftarrow DL[c \cup p, b \cap q; c \cap \neg b](a) \leftarrow p(a) \lor \neg q(a) \leftarrow p(a)$$
Outline

I. Background and Motivation

II. DL-Programs

III. Well-Supported Models

IV. Well-Supported Answer Set Semantics

V. Related Work

VI. Summary and Future Work
Fages’ Well-Supportedness

- Fages’ well-supportedness condition (Fages 1994, JMLCS):
  A model $I$ of a normal logic program is well-supported if there is a level mapping on $I$ such that for every $a \in I$, there is a rule
  
  $$ a \leftarrow A_1, \ldots, A_m, \text{not } B_1, \ldots, \text{not } B_n $$

  where $I$ satisfies the rule body and the level of each $A_i$ is below the level of $a$.

- This well-supportedness condition does not apply to DL-programs, due to occurrences of dl-atoms.
To handle dl-atoms, we introduce **up to satisfaction**.

Informally, for \( E \subseteq I \subseteq HB_R \),

\[(E, I) \models_L \alpha \text{ if for every } F \text{ with } E \subseteq F \subseteq I, F \models_L \alpha.\]

\((E, I) \models_L \alpha\) implies that the truth of \( \alpha \) depends only on \( E \) and \( I^- \), and is independent of \( I \setminus E \).

For instance, if \( E = \{a\}, I = \{a, b, c\} \) and \( \alpha = a \land \neg d \), then for every \( F \) with \( E \subseteq F \subseteq I \), \( F \models_L \alpha \). Therefore,

\[(E, I) \models_L \alpha.\]
up to Satisfaction \((E,I) \models_L A\)

**Definition**  Let \(KB = (L,R)\) and \(E \subseteq I \subseteq HB_R\). For any ground literal \(A\), define *\(E\) up to \(I\) satisfies \(A\) under \(L\)*, denoted \((E,I) \models_L A\), as follows:

1. For a ground atom \(a \in HB_R\),
   \[ (E,I) \models_L a \text{ if } a \in E; \quad (E,I) \models_L \neg a \text{ if } a \notin I. \]

2. For a ground dl-atom \(A\),
   \[ (E,I) \models_L A \text{ if for every } F \text{ with } E \subseteq F \subseteq I, \ F \models_L A; \]
   \[ (E,I) \models_L \neg A \text{ if for no } F \text{ with } E \subseteq F \subseteq I, \ F \models_L A. \]
Monotonicity of $(E, I) \models_L A$

- **Proposition** Let $A$ be a ground atom or dl-atom. For any $E_1 \subseteq E_2 \subseteq I$,
  - if $(E_1, I) \models_L A$ then $(E_2, I) \models_L A$;
  - and if $(E_1, I) \models_L \text{not } A$ then $(E_2, I) \models_L \text{not } A$.

- We use this up to satisfaction to extend Fages’ well-supportedness condition and define well-supported models for DL-programs.
Well-Supported Models

- Informally, a model $I$ of a DL-program is strongly well-supported if there is a level mapping on $I$ such that for every $a \in I$, there is $E \subseteq I$ and a rule $a \leftarrow \text{body}(r)$, where $(E, I) \models_L \text{body}(r)$ and the level of each element in $E$ is below the level of $a$.

- Put another way,
  - $a \in I$ is supported by $\text{body}(r)$,
  - while the truth of $\text{body}(r)$ is determined by $E$ and $I^-$,
  - where no $b \in E$ is circularly dependent on $a$.

- This guarantees that strongly well-supported models are free of circular justifications.
Well-Supported Models

**Definition**  A model \( I \) of a DL-program \( KB = (L, R) \) is strongly well-supported if there exists a strict well-founded partial order \( \prec \) on \( I \) such that for every \( a \in I \), there is \( E \subset I \) and a rule \( a \leftarrow body(r) \) in \( ground(R) \) such that

\[
(E, I) \models_L body(r) \quad \text{and} \quad \text{for every } b \in E, \ b \prec a.
\]
Well-Supported Models

Example  Consider a DL-program $KB = (L, R)$, where $L = \emptyset$ and

$$R: \quad p(a) \leftarrow q(a)$$

$$q(a) \leftarrow DL[c \cup p, b \cap q; c \sqcup \neg b](a)$$

$I = \{p(a), q(b)\}$ is the only model of $KB$. It is also a weak, a strong, and an FLP answer set. However, $I$ is not a strongly well-supported model, since for $p(a) \in I$ there is no $E \subset I$ satisfying the well-supportedness condition.
Well-Supported Models

- **Theorem** Let $KB = (L, R)$ be a DL-program, where $L = \emptyset$ and $R$ is a normal logic program. A model $I$ is a strongly well-supported model of $KB$ iff $I$ is a well-supported model of $R$ under Fages’ definition.

- As a result, Fages’ well-supportedness condition is extended to DL-programs.
Outline

I. Background and Motivation

II. DL-Programs

III. Well-Supported Models

IV. Well-Supported Answer Set Semantics

V. Related Work

VI. Summary and Future Work
Consequence Operator $T_{KB}(E, I)$

- **Definition** Let $KB = (L, R)$ and $E \subseteq I \subseteq HB_R$. Define $T_{KB}(E, I) = \{a | a \leftarrow body(r) \in ground(R) \text{ and } (E, I) \models_L body(r)\}$

- **Monotonicity** property of $T_{KB}(E, I)$

**Theorem** Let $I$ be a model of $KB$. For any $E_1 \subseteq E_2 \subseteq I$, $T_{KB}(E_1, I) \subseteq T_{KB}(E_2, I) \subseteq I$. 
**Fixpoint** $T_{KB}^\alpha(\emptyset, I)$

- $T_{KB}^\alpha(\emptyset, I)$: a fixpoint from the monotone sequence
  
  $$\langle T_{KB}^i(\emptyset, I) \rangle_{i=0}^\infty$$
  
  with $T_{KB}^0(\emptyset, I) = \emptyset$ and $T_{KB}^{i+1}(\emptyset, I) = T_{KB}(T_{KB}^i(\emptyset, I), I)$

- **Theorem** Let $I$ be a model of $KB = (L, R)$. If $I = T_{KB}^\alpha(\emptyset, I)$ then $I$ is a minimal model of $KB$. 
Well-Supported Semantics

- **Definition** Let $I$ be a model of a DL-program $KB = (L, R)$. $I$ is an answer set of $KB$ if $I = T_{KB}^\alpha (\emptyset, I)$.

- Answer sets are exactly strongly well-supported models

**Theorem** $I$ is an answer set of $KB$ iff $I$ is a strongly well-supported model of $KB$.

- Therefore, we call such answer sets well-supported answer sets, which are free of circular justifications.
Well-Supported Semantics

Theorem  If $I$ is a well-supported answer set of $KB$, then

1. $I$ is a minimal model of $KB$.

2. $I$ is a strong answer set of $KB$ that is also a weak answer set of $KB$.

3. $I$ is an FLP answer set of $KB$. 
Outline

I. Background and Motivation

II. DL-Programs

III. Well-Supported Models

IV. Well-Supported Answer Set Semantics

V. Related Work

VI. Summary and Future Work
Related Work

1. Weak answer set semantics (Eiter et al. 2008, AIJ)
   - There are circular justifications by self-supporting loops.

2. Strong answer set semantics (Eiter et al. 2008, AIJ)
   - The problem of circular justifications persists.

3. FLP answer set semantics (Eiter et al. 2005, IJCAI)
   - Weak/strong answer sets may not be minimal models.
   - FLP answer sets are minimal models.
   - The problem of circular justifications persists.

4. Loop formula based semantics (Wang et al. 2010, TPLP)
   - The problem of circular justifications persists.
Related Work

- FLP answer set semantics is based on FLP-reduct, a concept introduced in (Faber et al. 2004, JELIA) to define answer set semantics for logic programs with aggregates.
- Our up to satisfaction relation is inspired by conditional satisfaction, a concept introduced in (Son et al. 2007, JAIR) to define answer set semantics for logic programs with aggregates.
- DL-programs and logic programs with aggregates are closely related. Exploiting the deep connection presents an interesting future work.
Outline

I. Background and Motivation
II. DL-Programs
III. Well-Supported Models
IV. Well-Supported Answer Set Semantics
V. Related Work
VI. Summary and Future Work
Summary and Future Work

- **Summary:**

  To resolve the semantic problem of circular justifications with DL-programs, we
  
  - extended Fages’ well-supportedness condition from normal logic programs to DL-programs, and
  
  - presented a well-supported semantics for DL-programs, which produces answer sets free of circular justifications.
Summary and Future Work

- Future work:
  - Extend the work to DL-programs with disjunctive rule heads.
  - Study the complexity properties.
  - Exploit the connection between DL-programs and logic programs with aggregates.
Thanks!

Yi-Dong Shen

ydshen@ios.ac.cn

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