Loop Formulas for Description Logic Programs

Yisong Wang\textsuperscript{1}, Jia-Huai You\textsuperscript{2}, Li-Yan Yuan\textsuperscript{2}, Yidong Shen\textsuperscript{3}

\textsuperscript{1} Guizhou University, China
\textsuperscript{2} University of Alberta, Canada
\textsuperscript{3} Institute of Software, Chinese Academy of Sciences, China

26th Int’l Conference on Logic Programming
1 Motivation

2 Preliminary
   - Description logic programs

3 Completion and Loop Formulas
   - Completion
   - Weak loop formulas
   - Strong loop formulas

4 Canonical Answer Sets

5 Related Work and Conclusions
Integration of (nonmonotonic) rules and ontologies/description logics

- **The Semantic Web**: provides meanings of information and services on WWW, based on a layered structure with rules on top of ontologies.

- **Default/Typicality** reasoning with ontologies

- **Natural Kinds**: e.g., What is a computer scientist?
Description Logics (DLs) and Answer Set Programming

Description Logics
- Logic formalisms for Ontologies by concepts, roles and individuals, and their relationships
- DLs are fragments of first-order logic, but have more efficient decision problems.
- Concepts can be composed (such as $\text{Student} \sqcap \text{Employed}$)
- A DL knowledge base is a set of inclusion axioms (such as $\text{partTimeStudent} \sqsubseteq \text{Student}$) and assertions (e.g. $C(a)$, $R(b, c)$).

Answer Set Programming
Rules of the form (sometime disjunctive rules)

$$A \leftarrow B_1, \ldots, B_m, \text{not } C_1, \ldots \text{not } C_n$$

where $A$, $B_i$, and $C$ are atoms.
Current Approaches

1. **The Tight Approach**: DL and rules in the same language, e.g. MKNF knowledge bases (Motik and Rosati 2010).

2. **The Hybrid Approach**: DL + hybrid rules (Rosati ’05, ’06), can be formulated as a variant of Quantified Equilibrium Logic (de Bruijn et al. ’07),

3. **The Loose Approach**: Description Logic Programs: a loose integration of answer set programming with description logics (Eiter et al. ’08)
A dl-program is a pair \((O, P)\), where \(O\) is a description logic knowledge base and \(P\) a finite set of rules of the form:

\[ A \leftarrow B_1, \ldots, B_m, \text{not } C_1, \ldots, \text{not } C_n \]

where \(A\) is an atom and each \(B_i\) or \(C_i\) is an atom or a dl-atom.

- Eiter et al. defined weak and strong answer sets; some weak answer sets have "self-supports".
- We discovered that some strong answer sets may also have self-supports.
- We study semantics of dl-programs by loop completion.
- Use loop formulas to characterize answer sets, particularly those that are free of self-supports.
Definitions

**DL-Atom:**

A *dl-atom* is an expression of the form

\[ DL[S_1 \ op_1 \ p_1, \ldots, S_m \ op_m \ p_m; Q](\vec{t}), \]

where

- each \( S_i \) is either a concept, a role or a special symbol in \( \{ \approx, \not\approx \} \); \( \text{op}_i \in \{ \oplus, \odot, \ominus \} \)
- \( p_i \) is a unary predicate symbol in \( \mathcal{P} \) if \( S_i \) is a concept, and a binary predicate symbol in \( \mathcal{P} \) otherwise.
- \( Q(\vec{t}) \) is a *dl-query*, i.e., a query on a concept, role, inclusion (e.g. \( C \sqsubseteq D \)), or their negations.
Examples

Example 1

\[ DL[\textit{Student} \oplus \textit{registered}; \textit{Student}](a) \]

Assuming, for any individual \( x \), if \( x \) is registered for a course (information outside ontology) then \( x \) is a student (\( x \) may not be a student in ontology), and we query if \( \text{Student}(a) \).

Example 2

\[ DL[\textit{Student} \ominus \textit{registered}; \neg \textit{Student} \sqcap \neg \textit{Employed}](a) \]

queries if \( [\neg \text{Student} \sqcap \neg \text{Employed}](a) \)
An interpretation $I$ of a dl-program $\mathcal{K} = (O, P)$ is a subset of $HB_P$ (the Herbrand base of $P$), which is a model of an atom or dl-atom $A$, denoted by $I \models_O A$:

- if $A \in HB_P$, then $I \models_O A$ iff $A \in I$;
- if $A$ is a dl-atom $DL(\lambda; Q)(\vec{t})$, then $I \models_O A$ iff $O(I; \lambda) \models Q(\vec{t})$ where $O(I; \lambda) = O \cup \bigcup_{i=1}^{m} A_i(I)$ and, for $1 \leq i \leq m$,

$$A_i(I) = \begin{cases} 
\{ S_i(\vec{e}) \mid p_i(\vec{e}) \in I \}, & \text{if } op_i = \oplus; \\
\{ \neg S_i(\vec{e}) \mid p_i(\vec{e}) \in I \}, & \text{if } op_i = \ominus; \\
\{ \neg S_i(\vec{e}) \mid p_i(\vec{e}) \not\in I \}, & \text{if } op_i = \ominus.
\end{cases}$$
Weak Answer Set

Notation: Donate a rule by $A \leftarrow Pos, not\ Neg$.

The weak dl-transform of a dl-program $\mathcal{K} = (O, P)$, relative to $O$ and an interpretation $I \subseteq HB_P$, denoted by $\mathcal{K}^{w,I}$, is the positive dl-program $(O, wP^I_O)$, where $wP^I_O$ is obtained from $P$ by deleting:

- the dl-rules s.t. either $I \not\models_O B$ for some dl-atom $B \in Pos$, or $I \models_O C$ for some $C \in Neg$; and
- the dl-atoms and not $A$ from the remaining dl-rules where $A$ is an atom or dl-atom.

$I$ is a weak answer set of $\mathcal{K}$ if $I$ is the least model of $\mathcal{K}^{w,I}$. 
Let $\mathcal{K} = (O, P)$ where

$$O = \{C \sqsubseteq G\} \text{ and } P = \{p(a) \leftarrow DL[C \oplus p; G](a)\}.$$ 

$\mathcal{K}$ has two weak answer sets, $M_1 = \emptyset$ and $M_2 = \{p(a)\}$.

**Imagine the situation:**

- $p(x)$: $x$ is accepted to the TPLP Special Issue of ICLP2010.
- $C(x)$: $x$ is an ICLP paper
- $G(x)$: $x$ is a good paper
- $a$ stands for "this paper"

The weak answer set $\{p(a)\}$ is "circularly justified".
The **strong dl-transform** of a dl-program $\mathcal{K} = (O, P)$, relative to $O$ and an interpretation $I \subseteq HB_P$, denoted by $\mathcal{K}^{s,I}$, is the positive dl-program $(O, sP^I_O)$, where $sP^I_O$ is obtained from $P$ by deleting:

- the dl-rules s.t. either $I \not\models_O B$ for some nonmonotonic $B \in Pos$, or $I \models_O C$ for some $C \in Neg$; and
- the nonmonotonic dl-atoms and *not* $A$ from the remaining dl-rules where $A$ is an atom or dl-atom.

$I$ is a **strong answer set** of $\mathcal{K}$ if it is the least model of $\mathcal{K}^{s,I}$.
Characterization of Weak and Strong Answer Sets

- The idea [Lin and Zhao 2002]: The answer sets of a normal program can be characterized by models of Clark’s predicate completion satisfying its loop formulas.
- ASP solvers ASSAT and CMODELS are built on this idea.
- Loop formulas have been defined for a number of extensions of normal programs.

The completion of a dl-program $\mathcal{K} = (O, P)$ is defined as usual by regarding dl-atoms as formulas. Then, the models of the completion of $\mathcal{K}$ are supported models.
The **weak positive dependency graph** of \( \mathcal{K} \), written \( G^w_K \), is the directed graph \((HB_P, E)\) where \((u, v) \in E\) if \( P \) has a dl-rule with the head \( A = u \) and \( v \in Pos \).

A nonempty subset \( L \) of \( HB_P \) is a **weak loop** of \( \mathcal{K} \) if there is a cycle in \( G^w_K \) which goes through only and all the nodes in \( L \).

The **weak loop formula** of a weak loop \( L \), written \( wLF(L, \mathcal{K}) \), is:

\[
\bigvee L \supset \bigvee_{1 \leq i \leq n} \left( \bigwedge_{B \in \text{Pos}_i} B \land \bigwedge_{C \in \text{Neg}_i} \neg C \right)
\]

where \((h_1 \leftarrow \text{Pos}_1, \text{not Neg}_1), \ldots, (h_n \leftarrow \text{Pos}_n, \text{not Neg}_n)\) are all the rules in \( P \) such that \( h_i \in L \) and \( \text{Pos}_i \cap L = \emptyset \).
The **strong positive dependency graph** of $\mathcal{K}$, denoted by $G^s_{\mathcal{K}}$, is the directed graph $(HB_P, E)$, where $(p(\vec{c}), q(\vec{c}')) \in E$ if $P$ has a rule with head $A = p(\vec{c})$ and, for some $B \in Pos$, either

1. $B = q(\vec{c}')$, or
2. $B$ is a monotonic dl-atom mentioning the predicate $q$ and $\vec{c}'$ is a tuple of constants matching the arity of $q$.

The **strong loop formula** of $L$, written $sLF(L, \mathcal{K})$, is:

$$\bigvee L \supset \bigvee_{1 \leq i \leq n} \left( \bigwedge_{B \in Pos_i} \gamma(B, L) \land \bigwedge_{C \in Neg_i} \neg C \right)$$

where $\gamma(Q, L) = IF(Q, L)$ if $Q$ is monotonic, and $Q$ otherwise.
Example

Consider the weak answer set \{p(a)\} of program \((\emptyset, P)\), where \(P\) is the single rule program

\[ p(a) \leftarrow DL[c \oplus p; c](a) \]

Let \(A = DL[c \oplus p; c](a)\) and \(L = \{p(a), p(b)\}\)

Then, \(IF(A, L)\) is the formula:

\[ DL[c \oplus p_L; c](a) \land (p_L(a) \leftrightarrow p(a) \land a \neq a) \land (p_L(b) \leftrightarrow p(b) \land a \neq b) \]

which is equivalent to

\[ DL[c \oplus p_L; c](a) \land \neg p_L(a) \land (p_L(b) \leftrightarrow p(b)) \]
Let $\mathcal{K} = (O, P)$ be a dl-program and $I \subseteq HB_P$.

**Theorem**

$I$ is a weak answer set of $\mathcal{K}$ iff $I \models_O \text{COMP}(P) \cup \text{wLF}(\mathcal{K})$ where $\text{wLF}(\mathcal{K})$ is the set of weak loop formulas of all weak loops of $\mathcal{K}$.

**Theorem**

$I$ is a strong answer set of $\mathcal{K}$ iff $I^* \models_O \text{COMP}(P) \cup \text{sLF}(\mathcal{K})$ such that $I^*$ is the extension of $I$ where $\text{sLF}(\mathcal{K})$ is the set of strong loop formulas of all strong loops of $\mathcal{K}$.

The above two theorems suggest an alternative approach to computing the weak and strong answer sets of dl-programs: SAT solvers + DL-reasoners.
Example

Let $\mathcal{K} = (\emptyset, P)$ where $P$ consists of

\[ p(a) \leftarrow \text{not} \ DL[c \ominus p; \neg c](a). \]

$\mathcal{K}$ has a strong answer set \{p(a)\} where p(a) is "justified" by:

\[ p(a) \Rightarrow \text{not} \ DL[c \ominus p; \neg c](a) \Rightarrow p(a). \]

- Weak answer sets allow self-supporting loops involving any dl-atoms (either monotonic or nonmonotonic);
- Strong answer sets allow self-supporting loops only involving nonmonotonic dl-atoms and their default negations.
The canonical dependency graph of a dl-program $\mathcal{K} = (O, P)$, written $G^c_{\mathcal{K}}$, is the directed graph $(HB_P, E)$, where $(u, v) \in E$ if there is a rule with head $A = u$ and there exists an interpretation $I \subseteq HB_P$ such that either:

1. $I \not\models_O B$ and $I \cup \{v\} \models_O B$, for some $B \in Pos$, $v$ is called a positive monotonic (resp., nonmonotonic) dependency of $B$ if $B$ is a monotonic (resp., nonmonotonic) dl-atom, or
2. $I \models_O B$ and $I \cup \{v\} \not\models_O B$, for some $B \in Neg$. $v$ is called a negative nonmonotonic dependency of $B$. 
Examples of Dependencies

Let $\mathcal{K} = (O, P)$ be a dl-program where $O = \emptyset$ and $P$ consists of

$p(a_1) \leftarrow DL[c \oplus p, c](a_1),$
$p(a_2) \leftarrow DL[c \oplus p, b \oplus q; c \land \neg b](a_2),$
$p(a_3) \leftarrow not DL[c \ominus p, \neg c](a_3),$
$p(a_4) \leftarrow p(a_4).$

The weak (strong and canonical) dependency of $\mathcal{K}$ is illustrated by the following figure

Figure: The positive dependency relations on $HB_P$
Example

Let $\mathcal{K} = (\emptyset, P)$ and $I = \{p(a)\}$ where

$$P = \{p(a) \leftarrow \text{not } DL[c \oplus p, \neg c](a)\}$$

Note that $I \models_o \text{COMP}(\mathcal{K})$. However $I$ is not a canonical answer set of $\mathcal{K}$ because $\mathcal{K}$ has a canonical loop $L = \{p(a)\}$ and $cLF(L, I, \mathcal{K})$ is equivalent to

$$p(a) \supset \neg DL[c \oplus p_L; \neg c](a) \land (p_L(a) \leftrightarrow p(a) \land (a \neq a))$$
Proposition

Let $\mathcal{K} = (O, P)$ be a dl-program.

1. If $I$ is a canonical answer set of $\mathcal{K}$ then $I$ is minimal in the sense that $\mathcal{K}$ has no canonical answer set $I'$ such that $I' \subseteq I$.

2. If $I \subseteq HB_P$ is a canonical answer set of $\mathcal{K}$ then $I$ is noncircular (the notion of ”circularity” extended from [Falimeri et al. ’05].

3. If $I \subseteq HB_P$ is a canonical answer set of $\mathcal{K}$ then $I$ is a strong answer set of $\mathcal{K}$.

4. If $P$ does not mention the operator $\ominus$ then $I \subseteq HB_P$ is a canonical answer set of $\mathcal{K}$ if and only if $I$ is a strong answer set of $\mathcal{K}$. 
The relationships among different proposals are not clear:

- It is known how to translate dl-programs (mentioning no $\ominus$) to MKNF knowledge bases, but it’s unclear how to deal with $\ominus$;
- A variant of Quantified Equilibrium Logic (QEL) captures the existing hybrid approaches, the relation between dl-programs and QEL or first-order theories is not known, thus it’s unclear whether any existing result on loop formulas (e.g. [Lee and Meng ’08]).
Conclusions

1. We generalized the notions of loops and loop formulas of ASP to dl-programs to characterize weak and strong answer sets.
2. We proposed a new characterization of answer sets for dl-programs – canonical answer sets which are free of circular justifications.
3. The proposed notion of loop formula reveals some essential differences among weak, strong and canonical answer sets of dl-programs.

Future Work

- Does there exist a definition of canonical answer set without using loop formulas?
- The complexity of the new semantics?
- How exactly can one build a system using a SAT solver and a DL-reasoner?