Normal Description Logic Programs as Default Theories

Yisong Wang  Jia-Huai You  Liyan Yuan  Yi-Dong Shen

Nonmon@30
Lexington, Kentucky
Applications of ASP/SAT fall into ...

1. **Stand alone** applications
2. **ASP/SAT embedded** in a larger application; e.g., Eclipse
3. **ASP as a host language**, with other special inference mechanisms embedded
   - Logic programs with constraints;
   - Integration of nonmonotonic rules with special theories (ontologies/description logics)
Integration of rules and ontologies/description logics

- Ontologies are knowledge bases; can be expressed by description logics in which one defines concepts, roles and individuals, and their relationships.
- Knowledge in an ontology can be enhanced by knowledge external to it
- Rules are commonly recognized forms of reasoning
- Default/Typicality reasoning with ontologies
- Natural Kinds; e.g., What is a scientist?
Current Approaches

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

2. **The Tight (Hybrid) Approach**: e.g., DL + hybrid rules (Rosati '05, '06), (de Bruijn et al. '07)

3. **The Loose Approach**: DL-Programs (Eiter et al. '08)

However,

- There is little insight on the relationships among different approaches.

- Default logic seems to be a natural candidate but its potential has not been explored.
Contributions

We formulate two translations from description logic programs (dl-programs) to default theories

1. To eliminate the constraint operator from a (normal) dl-program so that all dl-atoms become monotonic. (A dl-program is normal if there are no monotonic dl-atoms that mention the constraint operator ⊖).

2. To translate a dl-program to a default theory such that
   - if the dl-program is normal, combined with the first translation, the translated default theory preserves strong answer sets;
   - for an arbitrary dl-program, the default extensions of its default translation correspond to strong answer sets.
The Loose Approach

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:

\[
A \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n \tag{1}
\]

where \(A\) is an atom, each \(B_i\) is an atom or a **dl-atom**.

**Notation**

- \(C\): a finite set of constants, which is a subset of those in \(O\).
- \(\Sigma\): a finite set of predicate symbols disjoint from those in \(O\).
- \(HB_P\): the **Herbrand base** of \(P\), which is the set of atoms formed from the predicate symbols of \(\Sigma\) and the constants of \(C\). An **interpretation** is a subset of \(HB_P\). (Let's consider only those symbols appearing in \(P\).)
DL-Atom: A form of meta-level atom

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

\[
DL[S_1 \ op_1 \ p_1, \ldots, S_m \ op_m \ p_m; Q](\vec{t}), \ (m \geq 0) \tag{2}
\]

where

- each \(S_i\) is either a concept, a role or a special symbol in \(\{\approx, \not\approx\}\);
- \(op_i \in \{\oplus, \odot, \ominus\}; \ominus\) is called the *constraint operator*;
- \(p_i\) is a predicate symbol in \(\Sigma\) matching its arity to that of \(S_i\);
- \(Q(\vec{t})\) is a *dl-query*, i.e., it may query a membership of a concept, a role, an concept inclusion, an equality and their negations, etc.
An interpretation $I$ is a model of an atom or dl-atom $A$ under $O$, written $I \models_O A$, if the following holds:

- if $A \in HB_P$, then $I \models_O A$ iff $A \in I$;
- if $A$ is a dl-atom $DL(\lambda; Q)(\vec{t})$ of the form (2), 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 $1 \leq i \leq m$,

$$A_i(I) = \begin{cases} 
\{S_i(\vec{e})|p_i(\vec{e}) \in I\}, & \text{if } op_i = \oplus; \\
\{\neg S_i(\vec{e})|p_i(\vec{e}) \in I\}, & \text{if } op_i = \ominus; \\
\{\neg S_i(\vec{e})|p_i(\vec{e}) \notin I\}, & \text{if } op_i = \ominus;
\end{cases}$$

where $\vec{e}$ is a tuple of constants over $C$. 
Notation

\( DL_?^K \): the set of nonmonotonic dl-atoms in a dl-program \( DL^K \).

The **strong dl-transform** of \( K \) relative to \( O \) and an interpretation \( I \subseteq HB_P \), denoted \( K^{s,I} \), is the positive dl-program \( (O, sP^I_O) \), where \( sP^I_O \) is obtained from \( P \) by deleting:

- the dl-rule \( r \) of the form (1) s.t. either \( I \not\models_O B_i \) for some positive \( B_i \in DL_?^K \), or \( I \models_O B_j \) for some negative atom or dl-atom \( B_i \); and
- the nonmonotonic dl-atoms and **not** \( A \) from the remaining dl-rules where \( A \) is an atom or dl-atom.

An interpretation \( I \) is a **strong answer set** of \( K \) if it is the least model of \( K^{s,I} \).
Example

Consider $\mathcal{K} = (O, P)$ where $O = \emptyset$ and

$$P = \{p(a) \leftarrow DL[c \oplus p, b \ominus q; c \sqcap \neg b](a)\}$$

Both $\emptyset$ and $\{p(a)\}$ are strong answer sets of $\mathcal{K}$. 
Deleting constraint operator

Let $\mathcal{K} = (O, P)$ be a dl-program and $r \in P$ a dl-rule. We define $\pi(r)$ to be a set of rules as follows:

(i) if $r$ is of the form $(h \leftarrow \text{not } DL[\lambda; Q](\vec{t}))$, where $DL[\lambda; Q](\vec{t})$ is a nonmonotonic dl-atom, then $\pi(r)$ consists of rule (3) below and the instantiations of rules of the form (4):

\begin{align*}
    h & \leftarrow \text{not } DL[\lambda'; Q](\vec{t}), \\ 
    p'_i(\vec{X}_i) & \leftarrow \text{not } p_i(\vec{X}_i), \quad (1 \leq i \leq k)
\end{align*}

where $p_i$ are all the predicate symbols occurring in $\lambda$ in the form of $S_i \ominus p_i$ for some $S_i$, and $\lambda'$ is obtained from $\lambda$ by replacing “$S_i \ominus p_i$” with “$S_i \odot p'_i$”, where $p'_i$ is a fresh predicate symbol matching the arity of $p_i$;
(ii) if \( r \) is of the form \( (h \leftarrow DL[\lambda; Q](\vec{t})) \), where \( DL[\lambda; Q](\vec{t}) \) is a nonmonotonic dl-atom, then \( \pi(r) \) consists of

\[
\begin{align*}
    h & \leftarrow \text{not } h', \\
    h' & \leftarrow \text{not } DL[\lambda'; Q](\vec{t}),
\end{align*}
\]

and the instantiated rules obtained from (4), where \( \lambda' \) is the same as before and \( h' \) is a fresh atom.

(iii) otherwise, \( \pi(r) = \{r\} \).
Example

Consider again $\mathcal{K} = (\emptyset, P)$ where

$$P = \{ p(a) \leftarrow DL[c \oplus p, b \ominus q; c \cap \neg b](a) \}$$

$\pi(\mathcal{K}) = (\emptyset, \pi(P))$ where $\pi(P)$ is

$$\left\{ \begin{array}{l}
p(a) \leftarrow \neg h', \\
h' \leftarrow \neg DL[c \oplus p, b \odot q'; c \cap \neg b](a), \\
q'(a) \leftarrow \neg q(a)
\end{array} \right\}$$

$\pi(\mathcal{K})$ has exactly two strong answer sets, $\{ q'(a), h' \}$ and $\{ q'(a), p(a) \}$. When restricted to the language of $\mathcal{K}$, they are $\emptyset$ and $\{ p(a) \}$. 
Theorem

Let $\mathcal{K} = (O, P)$ be a dl-program. An interpretation $I \subseteq HB_P$ is a strong (resp., weak) answer set of $\mathcal{K}$ if and only if $\pi(\mathcal{K})$ has a strong (resp., weak) answer set $I^*$ such that $I = I^* \cap HB_P$. 
Let $\mathcal{K} = (O, P)$ be a dl-program. We define $\tau(\mathcal{K})$ to be the default theory $(D, W)$ as follows:

- $W$ is a first-order theory corresponding to $O$,
- $D$ consists of (1) for each atom $p(\vec{c}) \in HB_P$, the default
  \[
  \frac{\neg p(\vec{c})}{\neg p(\vec{c})},
  \]
  and (2) for each dl-rule $r$ of the form (1) of $P$, the default
  \[
  \bigwedge_{1 \leq i \leq m} \tau(B_i) : \neg \tau(B_{m+1}), \ldots, \neg \tau(B_n)
  \]
  where

\[
A
\]
\( \tau(A) \) is defined as

- if \( A \) is an atom \( \tau(A) = A \),
- if \( A \) is a dl-atom of the form (2) then \( \tau(A) \) is the first-order sentence:

\[
\left[ \bigwedge_{1 \leq i \leq n} \tau(S_i \ op_i \ p_i) \right] \supset Q(\vec{t}) \quad \text{where}
\]

\[
\tau(S \ op \ p) = \begin{cases} 
\bigwedge_{\vec{c} \in \vec{C}} [p(\vec{c}) \supset S(\vec{c})] & \text{if } op = \oplus \\
\bigwedge_{\vec{c} \in \vec{C}} [p(\vec{c}) \supset \neg S(\vec{c})] & \text{if } op = \ominus \\
\bigwedge_{\vec{c} \in \vec{C}} [\neg p(\vec{c}) \supset \neg S(\vec{c})] & \text{if } op = \ominus 
\end{cases}
\]

where \( \vec{c} \) is a tuple of constants over \( C \) matching the arity of \( p \) and we identify \( S(\vec{c}) \) with its corresponding first-order sentence.
Example

Consider again $\mathcal{K} = (\emptyset, P)$ where

$$P = \{ p(a) \leftarrow DL[c \oplus p, b \ominus q; c \sqcap \neg b](a) \}$$

$\tau(\mathcal{K}) = (\{d_1, d_2, d_3\}, \emptyset)$ where

$$d_1 = \frac{\neg p(a)}{\neg p(a)}, \quad d_2 = \frac{\neg q(a)}{\neg q(a)},$$

$$d_3 = \frac{(p(a) \supset c(a)) \land (\neg q(a) \supset \neg b(a)) \supset c(a) \land \neg b(a)}{p(a)}.$$

$Th(\{\neg p(a), \neg q(a)\})$ is the unique extension of $\tau(\mathcal{K})$ though we know that $\mathcal{K}$ has two strong answer set $\emptyset$ and $\{p(a)\}$. 
Theorem

Let $\mathcal{K} = (O, P)$ be a canonical dl-program and $I \subseteq HB_P$. If $O$ is consistent then $I$ is a strong answer set of $\mathcal{K}$ if and only if $E = Th(O \cup I \cup \neg I)$ is an extension of $\tau(\mathcal{K})$.

Theorem

Let $\mathcal{K} = (O, P)$ be a dl-program where $O$ is consistent. If a theory $E$ is an extension of $\tau(\mathcal{K})$ then $E \cap HB_P$ is a strong answer set of $\mathcal{K}$. 

In the above, we assumed $O$ is consistent. But a dl-program with an inconsistent DL knowledge base may have nontrivial strong answer sets.

Thus, we provide yet another translation to default logic, which preserves strong answer sets for normal dl-programs, even in the case that the given DL knowledge base is inconsistent.
(Eiter et al 2010) proposed a translation to eliminate the constraint operator, for the well-founded semantics. But that translation does not preserve strong answer sets.

(Motik and Rosati 2010) translated a dl-atom without the constraint operator to a first order sentence, and then to an MKNF KB and showed a one-to-one correspondence between strong answer sets of the former and MKNF models of the latter.
We present an approach to translating normal dl-programs to default theories that preserves strong answer sets, by

1. eliminating ⊕ from nonmonotonic dl-atoms, and then
2. translating dl-programs to default theories.

These results have several implications:

1. The first improves a result of (Motik and Rosati 2010)
2. The second translates arbitrary dl-programs to default theories whose extensions correspond to strong answer sets.
3. Default logic is a potential framework for integrating ontologies and rules.
The semantics of dl-programs have been re-defined by minimal models of FLP reduct on HEX programs. That however doesn’t seem to avoid self-supporting loops.

Example

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

\[
\begin{align*}
  r(a) &\leftarrow s(a) \\
  s(a) &\leftarrow p(a) \\
  p(a) &\leftarrow DL[S \oplus r, S' \ominus s, S \sqcup \neg S'](a)
\end{align*}
\]

Its only strong answer set is $\{r(a), s(a), p(a)\}$, and it is the FLP answer set of $K$. 