Reconciling Well-Founded Semantics of DL-Programs and Aggregate Programs

Jia-Huai You, John Morris, Yi Bi

University of Alberta

October 24, 2012

Interest in well-founded semantics for extensions of ASP

- Pelov's theory for aggregate programs (2001-2004), based on fixpoint operators on bi-lattices, developed in
  - Denecker, Marek,Truszczynski: Uniform semantic treatment of default and autoepistemic logics. Artif. Intell. 2003
  - Denecker, Marek, and Truszczynski. Ultimate approximation and its application in nonmonotonic knowledge representation systems. Inf. and Comput., 2004
- Well-founded semantics for DL-programs (Eiter et al. 2004,2010)
- Unfounded sets for aggregate programs (Faber 2005)

In this paper, we show that the latter two are special cases of Pelov's theory.

Pelov's Theory for Well-Founded and Stable Semantics

Program:

$$A \leftarrow F$$

where A is an (ordinary) atom and F is a formula composed of atoms and aggregate atoms.
For simplicity, let's consider sets of rules of the form

$$A \leftarrow B_1, ..., B_k, not \ \ C_1, ..., not \ \ C_m$$

where A is an atom and B_i and C_i are atoms or aggregates.

- Semantics is defined by fixpoint operators over bi-lattices.
- well-founded semantics by the least fixpoint
- stable models by 2-valued (maximal) fixpoints.

Some Background

Given a complete lattice $\langle L, \preceq \rangle$, the bilattice induced from it is the structure $\langle L^2, \preceq, \preceq_p \rangle$, where for all $x, y, x', y' \in L$,

$$
\begin{array}{ll}
(x, y) \preceq (x', y') & \text{if and only if } \; x \preceq x' \text{ and } y \preceq y' \\
(x, y) \preceq_p (x', y') & \text{if and only if } \; x \preceq x' \text{ and } y' \preceq y
\end{array}
$$

- $\preceq$ on L^2 is called the product order
- $\preceq_p$ is called the precision order, a complete lattice order on L^2.
- We are interested only in those pairs (x, y) that are consistent, i.e., $x \preceq y$. Denote the set of consistent pairs by L^c

3-valued Interpretation

- (x, y) can be viewed as a 3-valued interpretation.
- When $x = y$ it is said to be exact (2-valued).

Approximating Operator

To make the theory sufficiently general to cover all possible semantics, we allow quite arbitrary operators

Definition

Let $O : L \to L$ be an operator on a complete lattice $\langle L, \preceq \rangle$. We say that $A : L^c \to L^c$ is an approximating operator of O iff the following conditions are satisfied:

- A extends O, i.e., $A(x, x) = (O(x), O(x))$, for every $x \in L$.
- A is $\preceq_p$-monotone.

Operator A is only required to extend O on exact pairs, in addition to monotonicity

Least Fixpoints

For aggregate programs, given a language $\mathcal{L}_\Sigma$, a program Π, and a monotonic approximating operator A of some operator O, we compute a sequence

$$(\emptyset, \Sigma) = (u_0, v_0), (u_1, v_1), ..., (u_k, v_k), ..., (u_\infty, v_\infty)$$

where the internal $[u_i, v_i]$ is decreasing, i.e., $[u_{i+1}, v_{i+1}] \subset [u_i, v_i]$, for all i.

Well-founded fixpoint

It is computed by a stable revision operator using two component operators of A.

- $A^1(\cdot, b)$: A with b fixed
- $A^2(a, \cdot)$: A with a fixed

Give a pair (a, b), we compute a new lower estimate by

$$x_0 = \perp, x_1 = A^1(x_0, b), ..., x_{i+1} = A^1(x_i, b), ..., x_\infty = A^1(x_\infty, b)$$

and a new upper estimate by

$$y_0 = a, y_1 = A^2(a, y_0), ..., y_{i+1} = A^2(a, y_i), ..., y_\infty = A^2(a, y_\infty)$$

The standard immediate consequence operator extended to aggregate programs Π is:

$$\mathcal{T}_\Pi(I) = \{H(r) \mid r \in \Pi \text{ and } I \models B(r)\}. \tag{1}$$

To approximate $\mathcal{T}_\Pi$, Pelov et al. defined a three-valued immediate consequence operator Φ_Π^{aggr} for aggregate programs, parameterized by the choice of approximating aggregates, which maps 3-valued interpretations to 3-valued interpretations.

$$\Phi_\Pi^{aggr}(I_1, I_2) = (I_1', I_2')$$

from which two component operators are induced:

$$\Phi_\Pi^{aggr,1}(I_1, I_2) = I_1' \quad \text{and} \quad \Phi_\Pi^{aggr,2}(I_1, I_2) = I_2' \tag{2}$$

Ultimate Approximating Aggregate Relation

Son and Pontelli (2007) showed an equivalent definition of $\Phi_{\Pi}^{aggr,1}$ in terms of <u>conditional satisfaction</u>, when the approximating aggregate used is the <u>ultimate approximating aggregate</u>. In a similar way, an equivalent definition of $\Phi_{\Pi}^{aggr,2}$ can be obtained.

Definition

Let Π be an aggregate program, and I and M interpretations with $I \subseteq M \subseteq \Sigma$. Then,

$$\Phi_{\Pi}^{aggr,1}(I, M) = \{H(r) \mid r \in \Pi, \forall J \in [I, M], J \models B(r)\} \tag{3}$$

$$\Phi_{\Pi}^{aggr,2}(I, M) = \{H(r) \mid r \in \Pi, \exists J \in [I, M], J \models B(r)\} \tag{4}$$

The least fixpoint constructed by these two component operators iteratively above is called the <u>ultimate well-founded semantics</u>.

Example

Consider aggregate program Π

$$\begin{aligned}
&p(-1).\\
&p(-2) \leftarrow sum_{\leq}(\{x \mid p(x)\}, 2).\\
&p(3) \leftarrow sum_{>}(\{x \mid p(x)\}, -4).\\
&p(-4) \leftarrow sum_{\leq}(\{x \mid p(x)\}, 0).
\end{aligned}$$

$$\begin{aligned}
&(\emptyset, \Sigma)\\
&\Rightarrow (\{p(-1)\}, \Sigma)\\
&\Rightarrow (\{\{p(-1), p(-2), p(-4)\}, \Sigma)\\
&\Rightarrow (\{\{p(-1), p(-2), p(-4)\}, \Sigma - \{p(3)\})
\end{aligned}$$

which is the well-founded fixpoint.

Unfounded Sets

We may define the well-founded semantics by the first principle of unfounded sets.

Definition

(Unfounded set) Let Π be an aggregate program and $I \subseteq Lit_P$ be consistent. A set $U \subseteq HB_\Pi$ is an unfounded set of Π relative to I iff

> *For every $a \in U$ and every rule $r \in P$ with $H(r) = a$, either for some $b \in B^+(r)$, it holds that $S^+ \not\models b$ for each consistent $S \subseteq Lit_\Pi$ with $I \cup \neg.U \subseteq S$, or for some $b \in B^-(r)$, it holds that $S^+ \models b$ for each consistent $S \subseteq Lit_\Pi$ with $I \cup \neg.U \subseteq S$.*

Theorem

Let Π be an aggregate program. The well-founded semantics of Π coincides with the ultimate well-founded semantics of Π.

Some others are instances of this formalism

- By a mapping of dl-atoms to aggregates, the well-founded semantics of Eiter et al. is a special case of the ultimate well-founded semantics of its translation.
- Faber (2005) defined a notion of unfounded sets, which is again an instance of Pelov's theory.
- Well-founded semantics for dl-programs with aggregates can be defined uniformly based on unfounded sets.

Example

Consider $KB = (L, P)$ with $L = \{Vip \sqsubseteq CR\}$, possibly plus some assertions of individuals, where CR stands for Customer-Record, and P containing

1. $purchase(X) \leftarrow purchase(X, Obj), item(Obj).$
2. $client(X) \leftarrow DL[CR \uplus purchase; CR](X).$
3. $imp\_client(X) \leftarrow DL[Vip](X).$
4. $imp\_client(X) \leftarrow client(X),$
   $sum_{\geq}(\{ Y | item(Obj), cost(Obj, Y),$
   $purchase(X, Obj)\}, 100).$
5. $discount(X) \leftarrow imp\_client(X).$
6. $promo\_offer(X) \leftarrow DL[CR \cup\!\!\!\!- imp\_client; CR](X),$
   $card_{=}(\{ Y | purchase(Y)\}, 0).$

Summary

- The intuitive definition of unfounded set and the resulting WFS can be seen as special cases of Pelov's theory, which provides a foundation for logic programs with external atoms.
- The least fixpoint can be pre-computed to simplify programs for the purpose of computing answer sets (2-valued maximal fixpoints).
- Future work: the class of aggregate/dl-programs whose WFS can be computed in polynomial time.