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

Jia-Huai You, John Morris, Yi Bi

University of Alberta

October 24, 2012
Pelov’s theory for aggregate programs (2001-2004), based on fixpoint operators on bi-lattices, developed in


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.**
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

\[ A \leftarrow B_1, \ldots, B_k, \text{not } C_1, \ldots, \text{not } C_m \]

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

\[
\begin{align*}
(x, y) \leq (x', y') & \quad \text{if and only if } x \leq x' \text{ and } y \leq y' \\
(x, y) \leq_p (x', y') & \quad \text{if and only if } x \leq x' \text{ and } y' \leq y
\end{align*}
\]

- $\leq$ on $L^2$ is called the **product order**
- $\leq_p$ 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., $x \leq y$. 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.
For aggregate programs, given a language $\mathcal{L}_\Sigma$, a program $\Pi$, and a monotonic approximating operator $A$ of some operator $O$, we compute a sequence

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

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

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

and a new upper estimate by

$$y_0 = a, y_1 = A^2(a, y_0), \ldots, y_{i+1} = A^2(a, y_i), \ldots, y_\infty = A^2(a, y_\infty)$$
The standard immediate consequence operator extended to aggregate programs $\Pi$ is:

$$T_\Pi(l) = \{ H(r) \mid r \in \Pi \text{ and } l \models B(r) \}.$$  \hfill (1)

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

$$\Phi_\Pi^{aggr}(l_1, l_2) = (l'_1, l'_2)$$

from which two component operators are induced:

$$\Phi_\Pi^{aggr,1}(l_1, l_2) = l'_1 \quad \text{and} \quad \Phi_\Pi^{aggr,2}(l_1, l_2) = l'_2 \quad (2)$$
Son and Pontelli (2007) showed an equivalent definition of $\Phi_{\Pi}^{aggr,1}$ in terms of conditional satisfaction, when the approximating aggregate used is the ultimate approximating aggregate. In a similar way, an equivalent definition of $\Phi_{\Pi}^{aggr,2}$ can be obtained.

**Definition**

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

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

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

The least fixpoint constructed by these two component operators iteratively above is called the ultimate well-founded semantics.
Consider aggregate program $\Pi$

\[ p(-1). \]
\[ p(-2) \leftarrow \text{sum}_{\leq}(\{x \mid p(x)\}, 2). \]
\[ p(3) \leftarrow \text{sum}_{>}(\{x \mid p(x)\}, -4). \]
\[ p(-4) \leftarrow \text{sum}_{\leq}(\{x \mid p(x)\}, 0). \]

\[
(\emptyset, \Sigma) \\
\implies (\{p(-1)\}, \Sigma) \\
\implies (\{\{p(-1), p(-2), p(-4)\}, \Sigma) \\
\implies (\{\{p(-1), p(-2), p(-4)\}, \Sigma - \{p(3)\})
\]

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 $\Pi$ be an aggregate program and $I \subseteq \text{Lit}_P$ be consistent. A set $U \subseteq H_{\Pi}$ is an unfounded set of $\Pi$ 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 $S \subseteq \text{Lit}_\Pi$ with $I \cup \neg \cdot U \subseteq S$, or for some $b \in B^-(r)$, it holds that $S^+ \models b$ for each consistent $S \subseteq \text{Lit}_\Pi$ with $I \cup \neg \cdot U \subseteq S$. 
Theorem

Let $\Pi$ be an aggregate program. The well-founded semantics of $\Pi$ coincides with the ultimate well-founded semantics of $\Pi$. 
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.
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. $\text{purchase}(X) \leftarrow \text{purchase}(X, Obj), \text{item}(Obj)$.
2. $\text{client}(X) \leftarrow DL[CR \uplus \text{purchase}; CR](X)$.
3. $\text{imp\_client}(X) \leftarrow DL[Vip](X)$.
4. $\text{imp\_client}(X) \leftarrow \text{client}(X)$,
   $$\text{sum} \geq (\{ Y | \text{item}(Obj), \text{cost}(Obj, Y),
   \text{purchase}(X, Obj) \}, 100)$$.
5. $\text{discount}(X) \leftarrow \text{imp\_client}(X)$.
6. $\text{promo\_offer}(X) \leftarrow DL[CR \uplus \text{imp\_client}; CR](X),
   \text{card} \geq (\{ Y | \text{purchase}(Y) \}, 0)$.
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.