A Generalized Gelfond-Lifschitz Transformation for Logic Programs with Abstract Constraints

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

Jia-Huai You
University of Alberta
Edmonton, Canada
you@cs.ualberta.ca
http://www.cs.ualberta.ca/~you

AAAI 2007, Vancouver, Canada
Background

• **Answer Set Programming (ASP)**
  - Logic programming with the stable model semantics; an effective formalism for solving combinatorial search problems

• **Logic Programs with Abstract Constraints**
  - Extensions of ASP with means to model aggregate constraints in particular, and *abstract constraints on sets* in general
  - Represent and *reason with sets of atoms*, in contrast with traditional logic programs primarily for *reasoning with individuals* (Marek & Remmel 2004; Marek & Truszczynski 2004)
Abstract Constraint Atoms (C-Atoms)

- A c-atom $A = (Ad, Ac)$, where $Ad$ is a finite set of atoms and $Ac \subseteq 2^{Ad}$
  (Marek & Remmel 2004; Marek & Truszczynski 2004)
- Represent any constraints with a finite set $Ac$ of admissible solutions over a finite domain $Ad$

Logic Programs with C-atoms

- Consist of clauses of the form
  
  $H_1 \lor \ldots \lor H_k \leftarrow A_1, \ldots, A_m, \text{not } B_1, \ldots, \text{not } B_n$
  
  where $H_i, A_i$ and $B_i$ are either atoms or c-atoms
Issues of Semantics

- The Standard Gelfond-Lifschitz Transformation
  - For logic programs without c-atoms (Gelfond & Lifschitz 1988; 1991)
  - Not applicable to logic programs with c-atoms

- A Challenging Question:
  - What is an appropriate semantics for logic programs with c-atoms?
Existing Proposals

- **Unfolding (Translation) Approaches**
  - Transform $P$ with c-atoms to $P'$ without c-atoms and define an interpretation $I$ as a stable model of $P$ if it is a stable model of $P'$ (Pelov et al. 2003; Son et al. 2006)

- **Fixpoint (Operator-Based) Approaches**
  - Apply some immediate consequence operator to construct a fixed point $lfp(P)$ and define $I$ as a stable model if $I = lfp(P)$ (Marek & Truszczynski 2004; Pelov 2004; Son et al. 2006)

- **Minimal Model Approaches**
  - Define a stable model to be a minimal model (Faber et al. 2004)
Our Proposal

- Define the stable model semantics for logic programs with abstract constraints by developing A generalized Gelfond-Lifschitz transformation
Our Contributions

- **A Formal Definition of the Semantics of C-Atoms**
  - Currently, the meaning of a c-atom is interpreted by means of propositional interpretations (truth assignments)

- **A Succinct Abstract Representation of C-Atoms**
  - A c-atom is coded with a substantially smaller size than using the current power set form representation

- **A Generalization of the Gelfond-Lifschitz Transformation**
  - Used to define the stable model semantics for disjunctive logic programs with arbitrary c-atoms appearing anywhere in a clause
1. Semantics of C-Atoms

- **Marek & Truszczynski’s Definition**
  - The meaning of a c-atom $A$ is interpreted by means of propositional interpretations (truth assignments)
  - An interpretation $I$ satisfies $A = (Ad, Ac)$, written as $I \models A$, if $I \cap Ad \in Ac$; $I$ satisfies $\text{not } A$ if $I \cap Ad \notin Ac$

- **Our Observation**
  - Marek & Truszczynski’s truth assignment-based interpretation can be concisely formalized using a logic expression, thus leading to a formal definition of the semantics of c-atoms
1. Semantics of C-Atoms

- Our Formalization

**Definition 1** Let $A = (A_d, A_c)$ be a c-atom. Its semantics is defined by

$$A \equiv \bigvee_{S \in A_c} S \land \text{not} \ (A_d \setminus S)$$

(1)

$A = (\{a, b\}, \{\{a\}, \{b\}, \{a, b\}\})$

**Semantic definition**

$$A \equiv (a \land \text{not} \ b) \lor (b \land \text{not} \ a) \lor (a \land b)$$
1. Semantics of C-Atoms

- Justification of Our Formalization

**Theorem 1** An interpretation $I$ satisfies $A$ iff $I$ satisfies

$$\forall S \in A_c S \land \text{not} (A_d \setminus S)$$

$I$ satisfies $\text{not} A$ iff $I$ satisfies

$$\text{not} (\forall S \in A_c S' \land \text{not} (A_d \setminus S'))$$
Logical Equivalence Simplification

For any $S_1$ and $S_2$,

$$(S_1 \land \underline{L} \land S_2) \lor (S_1 \land \underline{not} \ L \land S_2) \equiv S_1 \land S_2$$

$$A = (\{a, b\}, \{\{a\}, \{b\}, \{a, b\}\})$$

**semantic definition**

$$A \equiv (a \land \text{not } b) \lor (b \land \text{not } a) \lor (a \land b)$$

**logically simplified**

$$A \equiv a \lor b$$
2. Abstract Representation of C-Atoms

- **Current Power Set Form Representation**
  - \( A = (Ad, Ac) \)
  - \( Ac \subseteq 2^{Ad} \) would be extremely large

- **Our Power Set Free Abstract Representation**
  - \( A = (Ad, Ac^*) \)
  - \( W \cup V \) in \( Ac^* \) covers all \( W \)-prefixed power sets of \( V \) in \( Ac \)
    
    i.e., \( W \cup V = \{ W \cup S | S \in 2^V \} \)
2. Abstract Representation of C-Atoms

\[ A_c = \{\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{c, d\}, \{a, b, c\}, \{b, c, d\}\} \]

Power set form representation

\[ A_c^* = \{\emptyset \uplus \{b, c\}, \{c\} \uplus \{a, b\}, \{c\} \uplus \{b, d\}\} \]

Abstract representation
2. Abstract Representation of C-Atoms

\[ A_c = \{\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{c, d\}, \{a, b, c\}, \{b, c, d\}\} \]

**W \uplus V covers a set S if \( W \subseteq S \) and \( S \subseteq (W \cup V) \)**
2. Abstract Representation of C-Atoms

\[ A_c = \{\emptyset, \{b\}, \{c\}, \{a, c\}, \{b, c\}, \{c, d\}, \{a, b, c\}, \{b, c, d\}\} \]

\[ A_c^* = \{\emptyset \cup \{b, c\}, \{c\} \cup \{a, b\}, \{c\} \cup \{b, d\}\} \]

\[ \{c\} \cup S, \quad S \in 2^{\{a, b\}} \]
2. Abstract Representation of C-Atoms

**Theorem 2** Let $A = (A_d, A_c)$ be a c-atom.

1. $A$ has a unique abstract form $(A_d, A_c^*)$.

2. An interpretation $I \models A$ iff $A_c^*$ contains $W \cup V$ covering $I \cap A_d$.

3. $A_c^*$ is power set free.

***** In many cases, $|A_c^*| \ll |A_c|$; in an extreme case, $|A_c|$ = $2^{|A_d|}$, but $|A_c^*| = 1$ ($A_c = 2^{A_d}$, $A_c^* = \{\emptyset \cup A_d\}$)
2. Abstract Representation of C-Atoms

- C-atoms can be characterized in terms of the abstract representation.

**Theorem 3** Let \( A \) be a c-atom. Then

\[
A \equiv \bigvee_{W \uplus V \in A^*_c} W \land \neg (A_d \setminus (W \cup V))
\]

***** This theorem lays a solid basis for the development of the semantics of logic programs with c-atoms.
2. Abstract Representation of C-Atoms

- Abstract Satisfiable Sets

**Definition 2** Let $A$ be a c-atom and $I$ an interpretation.

1. $W \cup V \in A^*_c$ is an *abstract satisfiable set* if $W \cup V$ covers $I \cap A_d$.

2. $W$ is called a *satisfiable set* if there is an abstract satisfiable set $W \cup V$. 
2. Abstract Representation of C-Atoms

- Characterizing C-Atoms in terms of Abstract Satisfiable Sets

**Theorem 4** Let $A$ be a c-atom and $I$ an interpretation. $I \models A$ iff $I$ satisfies

$$\forall \text{ each abs. sat. set } W \cup V \quad W \land \neg (A_d \setminus (W \cup V))$$
3. A Generalization of the Gelfond-Lifschitz Transformation

- **Key Ideas (1):** for each c-atom $A$ in the **body** of a clause

  \[
  \cdots \leftarrow \cdots, A, \cdots
  \]
  
  replaced by
  
  $\Theta_A$
  
  defined by
  
  $\Theta_A \leftarrow W$, for each abstract satisfiable set $W \cup \mathcal{V}$
3. A Generalization of the Gelfond-Lifschitz Transformation

- **Key Ideas (2):** for each c-atom $A$ in the head of a clause

  ... $A$ ... $\leftarrow$ ...

  replaced by

  $\beta_A$

  defined by

  $B \leftarrow \beta_A$, for each $B$ in $I \cap Ad$
  $\bot \leftarrow B, \beta_A$, for each $B$ in $Ad \setminus (I \cap Ad)$
  $\beta_A \leftarrow I \cap Ad$

**These new clauses define that $\beta_A$ iff $I \cap Ad$**
3. A Generalization of the Gelfond-Lifschitz Transformation

- **Key Ideas (3):** for a c-atom $A = (A_d, A_c)$, its negation $\neg A$ is treated as the complement of $A$; i.e.,

$$\neg A = (A_d, 2^{A_d} \setminus A_c)$$

the complement of $A_c$
Definition 3 Given a logic program $P$ and an interpretation $I$, the generalized Gelfond-Lifschitz transformation of $P$ w.r.t. $I$, written as $P^I$, is obtained from $P$ by performing the following four operations:

1. Remove from $P$ all clauses whose bodies contain either a negative literal $\text{not } A$ such that $I \nvdash \text{not } A$ or a c-atom $A$ such that $I \nvdash A$.

2. Remove from the remaining clauses all negative literals, and then

3. Replace each c-atom $A$ in the body of a clause with a special atom $\theta_A$ and introduce a new clause $\theta_A \leftarrow A_1, \ldots, A_m$ for each satisfiable set $\{A_1, \ldots, A_m\}$ of $A$ w.r.t. $I \cap A_d$.

4. Replace each c-atom $A$ in the head of a clause with $\bot$ if $I \nvdash A$, or replace it with a special atom $\beta_A$ and introduce a new clause $B \leftarrow \beta_A$ for each $B \in I \cap A_d$, a new clause $\bot \leftarrow B, \beta_A$ for each $B \in A_d \setminus (I \cap A_d)$, and a new clause $\beta_A \leftarrow I \cap A_d$. 
Stable Models under the Generalized Gelfond-Lifschitz Transformation

**Definition 4** For any logic program $P$, an interpretation $I$ is a *stable model* of $P$ if $I = M \setminus \{\theta_X, \beta_X\}$, where $M$ is a minimal model of the generalized Gelfond-Lifschitz transformation $P^I$. 
Main Properties (1)

**Theorem 5** Let $P$ be a logic program such that c-atoms appearing in the heads of its clauses are all elementary. Any stable model of $P$ is a minimal model of $P$.

** An elementary c-atom is of the form ($\{a\}$, {{a}}), where $a$ is an atom.
Main Properties (2)

**Theorem 6** Let $P$ be a non-disjunctive logic program. An interpretation $I$ is a stable model if and only if it is a stable model under Son et al.’s fixpoint definition.

Complexity

**Theorem 8** Let $P$ be a logic program with $n$ different $c$-atoms.

1. The time complexity of computing all satisfiable sets of $A$ is linear in the size of $A^*_c$.

2. The time complexity of the generalized Gelfond-Lifschitz transformation is bounded by $O(|P| + n \times (2M^*_{A_c} + M_{A_d} + 1))$, where $M^*_{A_c}$ and $M_{A_d}$ are the maximum sizes of $A^*_c$ and $A_d$ of a $c$-atom in $P$, respectively.

3. The size of $P^I$ is bounded by $O(|P| + n \times (M^*_{A_c} + M_{A_d} + 1))$.

4. The time to compute $A^*_c$ from $A_c$ is bounded by $O(|A_c|^3 \times |A_d|)$. 
Relationship to Existing Approaches

- Essentially different from the existing approaches in that we define the stable model semantics for logic programs with c-atoms by developing a generalized Gelfond-Lifschitz transformation based on the formal semantics and abstract representation of c-atoms.
Relationship to Existing Approaches (1)

- Let \( r \) be a clause \( B \leftarrow A_1, ..., A_m \). An unfolding approach (Pelov et al. 2003; Son and Pontelli 2006) will transform \( r \) into \( n_1 \ast ... \ast n_m \) new clauses of the form \( B \leftarrow \overline{A}_1, ..., \overline{A}_m \), where each \( \overline{A}_i \) is built from an aggregate solution of \( A_i \). Our approach transforms \( r \) into \( 1 + n'_1 + ... + n'_m \) clauses, where \( n'_i \) is the number of satisfiable sets of \( A_i \). In general, for each \( i \) we have \( n_i \gg n'_i \).

**\( n_i \) is the number of aggregate solutions of \( A_i \)**
Relationship toExisting Approaches (2)

- Stable models defined using our approach coincide with those applying Son et al.’s fixpoint approach (Son et al. 2006; 2007) for non-disjunctive logic programs with arbitrary c-atoms.

** Son et al. show that their fixpoint semantics coincides with that of Marek and Truszczyński (2004) for non-disjunctive logic programs with monotone c-atoms; with that of Faber et al. (2004) and Ferraris (2005) for positive basic logic programs with monotone c-atoms; with that of Denecker et al. (2001; 2003) for positive basic logic programs with arbitrary c-atoms.
Relationship to Existing Approaches (3)

- Our approach has the minimality property for the class of logic programs in which c-atoms appearing in clause heads are all elementary. It is different from the minimal model approach by Faber et al. (2004).
Summary

- We introduced a formal characterization of the semantics of c-atoms
- We created an abstract representation of c-atoms
- We developed a generalized Gelfond-Lifschitz transformation based on the formal semantics and abstract representation of c-atoms
- Stable models coincide with Son et al.'s fixpoint approach for non-disjunctive logic programs with arbitrary c-atoms
Thanks!

Yi-Dong Shen
ydshen@ios.ac.cn
http://lcs.ios.ac.cn/~ydshen/Shen-AAAAI07.pdf