Generality Relations in Answer Set Programming

Katsumi Inoue
National Institute of Informatics
Chiaki Sakama
Wakayama University

ICLP 2006. 08. 18

Comparing the Amounts of Information between Programs

- ⌘ Assessment of relative value of each theory
  - ⌘ **Generality**/Specificity and Informativeness
  - ⌘ **Equivalence** and Non-equivalence
  - ⌘ **Strength**/Weakness and Priority/Utility
- ⌘ Theory of generality is central in **Inductive Logic Programming** (ILP), in which domain-independent criteria to compose better theories are investigated.
- ⌘ Synthesizing a common generalized/specialized program from different sources of information is important in **Multi-Agent Systems** (MAS).

Comparing First-order Theories

- ⌘ T1, T2: first-order theory/program
- ⌘ T1 is **more general** than T2 if $T1 \models T2$ [Plotkin; Nilbet].
- ⌘ T1 and T2 are **logically equivalent** if $T1 \equiv T2$, i.e., $T1 \models T2$ and $T2 \models T1$.
- ⌘ Logically equivalent programs belong to the same equivalence class of the generality relation.

Comparing Nonmonotonic Theories

- P1, P2: logic programs with negation as failure
- When can we say that P1 is **more general** than (or is **more informative** than) P2?
- P1 and P2 are **equivalent** if P1 and P2 have the same semantical meaning:
  - *weak/strong equivalence* [Maher; Lifschitz et al.]
- Under which generality relation do equivalent programs belong to the same equivalence class?

Comparing Nonmonotonic Programs

⌘ **Example**:

P1 : p $\leftarrow$ ***not*** q

P2 : p $\leftarrow$ ***not*** q, q $\leftarrow$ ***not*** p,

P1 has the answer set: {p}
P2 has the answer sets: {p}, {q}

⌘ P1 is *more informative* than P2 in the sense that P1 has the **skeptical** consequences {p} which includes {}.

⌘ P2 is *more informative* than P1 in the sense that P2 has the **credulous** consequences {p,q} which includes {p}.

⌘ Thus, several generality measures can be considered.

Goals

- ⌘ We construct multiple criteria to decide if a program is more general than another program in **answer set programming**.
- ⌘ Generality relations are mathematically defined as **pre-orders** based on **comparing sets of answer sets**.
- ⌘ Any pair of programs should have both **minimal upper and maximal lower bounds** under such generality orderings.
- ⌘ We devise those generality orderings in such a way that any pair of **equivalent programs belong to the same equivalence class** that is induced from such pre-ordered sets.
- ⌘ We also provide the notion of **strong generality** that implies strong equivalence within the same equivalence class.

Extended Disjunctive Programs

- ⌘ Rule ***r*** :

  $L_1 \,;\, \ldots ;\, L_l \leftarrow L_{l+1}, \ldots, L_m, \textbf{\textit{not}}\ L_{m+1}, \ldots, \textbf{\textit{not}}\ L_n$

- ⌘ head(***r***)=$\{L_1, \ldots, L_l\}$,

  body$^+$(***r***)=$\{L_{l+1}, \ldots, L_m\}$, body$^-$(***r***)=$\{L_{m+1}, \ldots, L_n\}$.

- ⌘ Disjunctive rule: $l \geq 2$.
- ⌘ Integrity constraint (**IC**): $l = 0$.
- ⌘ NAF-free rule: $m = n$.
- ⌘ Fact: $l = m = n \geq 1$.
- ⌘ Extended logic program (**ELP**): $\forall$ ***r***: $l \leq 1$.

Answer Sets [Gelfond & Lifschitz]

I. When P is an NAF-free EDP,

S is an **answer set** of P if ***S*** is a minimal set satisfying:

1. For each ground rule ***r*** from P :

$$L_1 ; \ldots ; L_l \leftarrow L_{l+1} , \ldots, L_m ,$$

$$\{L_{l+1} , \ldots, L_m\} \subseteq \boldsymbol{S} \text{ implies } \{L_1 , \ldots, L_l\} \cap \boldsymbol{S} \neq \phi ;$$

2. If ***S*** contains a pair of complementary literals, then ***S*** = ***Lit***.

II. When P is any EDP,

the NAF-free EDP $P^{\boldsymbol{S}}$ is obtained as follows:

A rule

$$L_1 ; \ldots ; L_l \leftarrow L_{l+1} , \ldots, L_m$$

is in $P^{\boldsymbol{S}}$ iff there is a ground rule from P of the form:

$$L_1 ; \ldots ; L_l \leftarrow L_{l+1} , \ldots, L_m , \textbf{not } L_{m+1} , \ldots, \textbf{not } L_n$$

such that $\{L_{m+1} , \ldots, L_n\} \cap \boldsymbol{S} = \phi$.

S is an **answer set** of P if ***S*** is an answer set of $P^{\boldsymbol{S}}$.

Answer Set Semantics

- An answer set is **consistent** if it is not ***Lit***.
- A program is **consistent** if it has a consistent answer set; otherwise, **inconsistent**.
- An inconsistent program is either
  - **contradictory** if it has the answer set ***Lit***, or
  - **incoherent** if it has no answer set.
- $A(P)$: the set of all answer sets of P .
- A literal L is a **skeptical / credulous consequence** of P if L belongs to all/some answer sets in $A(P)$.
  - $skp(P)$: the set of skeptical consequences of P
  - $crd(P)$: the set of credulous consequences of P

Equivalence between Programs

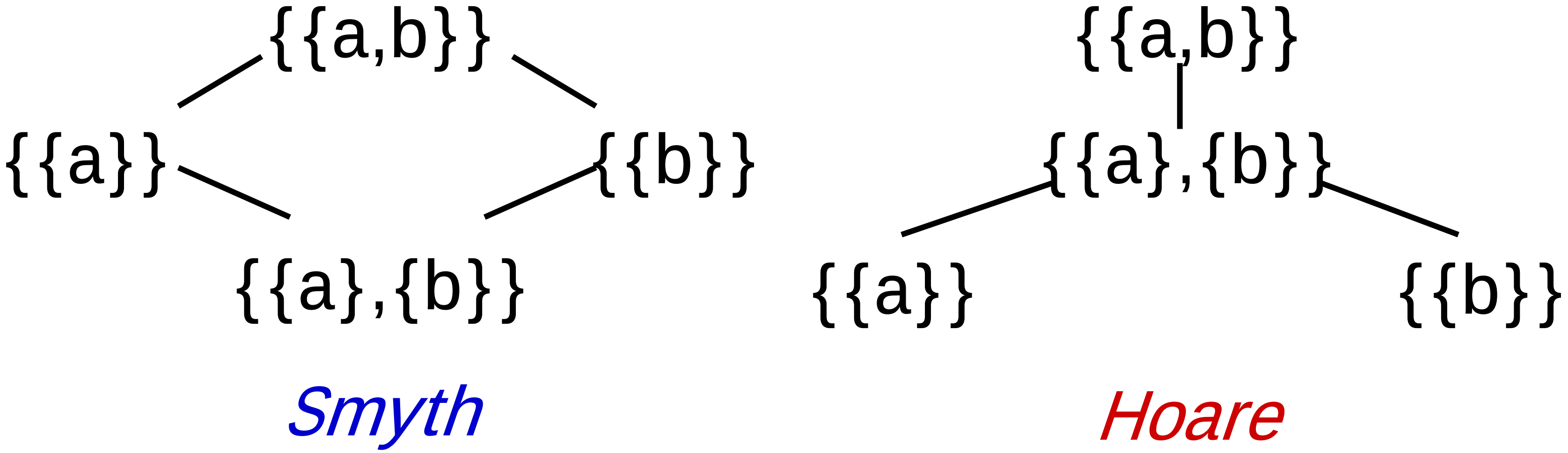
- Let P_1 , P_2 and R be programs.
- P_1 and P_2 are **(weakly) equivalent** if $A(P_1) = A(P_2)$.
- P_1 and P_2 are **strongly equivalent** [Lifschitz, Pearce & Valverde] if $A(P_1 \cup R) = A(P_2 \cup R)$ for any program R.
- Strong equivalence implies weak equivalence.

Ordering Answer Sets: Basic Intuition

⌘ In the FO case, $\{ a \wedge b \}$ is more informative than $\{ a \}$, which is more informative than $\{ a \vee b \}$.

In fact, $a \wedge b \models a \models a \vee b$.

⌘ In analogy, $\{\{a,b\}\} \geq \{\{a\}\} \geq \{\{a\},\{b\}\}$.



Ordering on Powersets

⌘ *pre-order* $\leq$: binary relation which is reflexive and transitive

⌘ *partial order* $\leq$: pre-order which is also anti-symmetric

⌘ $\langle D, \leq \rangle$: *pre-ordered set / poset*

⌘ S (D) : the power set of D

⌘ The **Smyth order**: for $X, Y \in$ S (D),

$$X \models^{\#} Y \text{ iff } \forall x \in X \, \exists y \in Y. \; y \leq x$$

⌘ The **Hoare order**: for $X, Y \in$ S (D),

$$X \models^{\flat} Y \text{ iff } \forall y \in Y \, \exists x \in X. \; y \leq x$$

⌘ Both $\langle$S $(D), \models^{\#} \rangle$ and $\langle$S $(D), \models^{\flat} \rangle$ are pre-ordered sets.

Ordering Logic Programs

⌘ $\langle \mathcal{S}(\boldsymbol{Lit}), \subseteq \rangle$: poset

⌘ $\mathcal{EDP}$: the class of all programs

⌘ $P, Q \in \mathcal{EDP}$

- P is **more #-general** than Q :

$$P \models^{\#} Q \quad \text{iff} \quad A(P) \models^{\#} A(Q)$$

- P is **more ♭-general** than Q :

$$P \models^{\flat} Q \quad \text{iff} \quad A(P) \models^{\flat} A(Q)$$

⌘**Theorem**: $P \models^{\#} Q$ and $Q \models^{\#} P$

iff $P \models^{\flat} Q$ and $Q \models^{\flat} P$

iff P and Q are weakly equivalent.

Ordering Logic Programs

⌘ **<u>Example</u>**:

P1 : $p \leftarrow \textbf{\textit{not}}\ q$

P2 : $p \leftarrow \textbf{\textit{not}}\ q, \quad q \leftarrow \textbf{\textit{not}}\ p,$

P3 : $p\ ;\ q \leftarrow$

P4 : $p \leftarrow \textbf{\textit{not}}\ \neg p, \quad q \leftarrow p$

$\mathcal{A}(P1) = \{\{p\}\}, \quad \mathcal{A}(P2) = \mathcal{A}(P3) = \{\{p\}, \{q\}\}, \quad \mathcal{A}(P4) = \{\{p,q\}\}$

- $P4 \models^{\#} P1 \models^{\#} P2$
- $P4 \models^{\flat} P2 \models^{\flat} P1$
- $P2 \models^{\#} P3 \models^{\#} P2, \quad P2 \models^{\flat} P3 \models^{\flat} P2$

Minimal Upper / Maximal Lower Bounds

- ⌘ Q is an *upper bound* of P1 and P2 in $\langle \mathcal{EDP}, \models^{\#/\flat} \rangle$ if Q $\models^{\#/\flat}$ P1 and Q $\models^{\#/\flat}$ P2.
- ⌘ An upper bound Q is an *mub* of P1 and P2 in $\langle \mathcal{EDP}, \models^{\#/\flat} \rangle$ if Q $\models^{\#/\flat}$ Q' implies Q' $\models^{\#/\flat}$ Q for any upper bound of P1 and P2.
- ⌘ Q is a *lower bound* of P1 and P2 in $\langle \mathcal{EDP}, \models^{\#/\flat} \rangle$ if P1 $\models^{\#/\flat}$ Q and P2 $\models^{\#/\flat}$ Q.
- ⌘ A lower bound Q is an *mlb* of P1 and P2 in $\langle \mathcal{EDP}, \models^{\#/\flat} \rangle$ if Q' $\models^{\#/\flat}$ Q implies Q $\models^{\#/\flat}$ Q' for any lower bound of P1 and P2.

Minimal Upper / Maximal Lower Bounds

- **Theorem**:
  - Q is an mub of P1 and P2 in $\langle \mathcal{EDP}, \models^{\#} \rangle$
    iff $A(Q) = min\{ S \uplus T \mid S \in A(P1), T \in A(P2) \}$,
    where $S \uplus T = S \cup T$, if consistent; ***Lit***, otherwise.
  - Q is an mlb of P1 and P2 in $\langle \mathcal{EDP}, \models^{\#} \rangle$
    iff $A(Q) = min(A(P1) \cup A(P2))$.
  - Q is an mub of P1 and P2 in $\langle \mathcal{EDP}, \models^{\flat} \rangle$
    iff $A(Q) = max(A(P1) \cup A(P2))$.
  - Q is an mlb of P1 and P2 in $\langle \mathcal{EDP}, \models^{\flat} \rangle$
    iff $A(Q) = max\{ S \cap T \mid S \in A(P1), T \in A(P2) \}$.
- A top / bottom element of $\langle \mathcal{EDP}, \models^{\#} \rangle$ is { p← ***not*** p } / {}.
- A top / bottom element of $\langle \mathcal{EDP}, \models^{\flat} \rangle$ is { p←, ¬p← } / { p← ***not*** p }.

Computing Mubs and Mlbs

- Given (answer sets of) P1 and P2, computation of an EDP Q such that
  - $\mathcal{A}(Q) = min\{\ S \uplus T \mid S \in \mathcal{A}(P1),\ T \in \mathcal{A}(P2)\ \}$
  - $\mathcal{A}(Q) = min(\ \mathcal{A}(P1) \cup \mathcal{A}(P2)\ )$
  - $\mathcal{A}(Q) = max(\ \mathcal{A}(P1) \cup \mathcal{A}(P2)\ )$
  - $\mathcal{A}(Q) = max\{\ S \cap T \mid S \in \mathcal{A}(P1),\ T \in \mathcal{A}(P2)\ \}$

  is considered as coordination/composition/consensus of programs in the context of multi-agent systems [Sakama & Inoue, 2004-2006].

- In general, such a program can also be constructed through the DNF-CNF translation.

Entailed Literals in More / Less General Programs

⌘ **Theorem**:

- If $P \models^{\#} Q$ then $skp(Q) \subseteq skp(P)$.
- If $P \models^{\flat} Q$ then $crd(Q) \subseteq crd(P)$.

⌘ Pre-orders based on skeptical/credulous entailment relations over sets of literals can also be defined.

⌘ **Theorem**:

An mub/mlb of P1 and P2 in $\langle \mathcal{EDP}, \models^{\#/\flat} \rangle$ is an mub/mlb of P1 and P2 in $\langle \mathcal{EDP}, \models_{skp/crd} \rangle$.

Strong Generality

⌘ $P, Q \in \mathcal{EDP}$

- P is **strongly more #-general** than Q :
  $P \trianglerighteq^{\#} Q$ iff $P \cup R \models^{\#} Q \cup R$ for any program R.
- P is **strongly more ♭-general** than Q :
  $P \trianglerighteq^{\flat} Q$ iff $P \cup R \models^{\flat} Q \cup R$ for any program R.

⌘ $P \trianglerighteq^{\#/\flat} Q$ implies $P \models^{\#/\flat} Q$.

⌘ $\langle \mathcal{EDP}, \trianglerighteq^{\#/\flat} \rangle$ is a pre-ordered set.

⌘**Theorem**: $P \trianglerighteq^{\#} Q$ and $Q \trianglerighteq^{\#} P$

iff $P \trianglerighteq^{\flat} Q$ and $Q \trianglerighteq^{\flat} P$

iff P and Q are strongly equivalent.

Strong Generality

⌘ **Example**:

$$P1: \quad p \leftarrow \textbf{\textit{not}}\ q$$

$$P2: \quad p \leftarrow \textbf{\textit{not}}\ q, \quad q \leftarrow \textbf{\textit{not}}\ p,$$

$$P3: \quad p\ ;\ q \leftarrow$$

$$P4: \quad p \leftarrow \textbf{\textit{not}}\ \neg p, \ \ q \leftarrow p$$

$\mathcal{A}(P1) = \{\{p\}\}, \quad \mathcal{A}(P2) = \mathcal{A}(P3) = \{\{p\}, \{q\}\}, \quad \mathcal{A}(P4) = \{\{p,q\}\}$

- $P1 \trianglerighteq^{\#} P2 \trianglerighteq^{\#} P3$
- $P3 \trianglerighteq^{\flat} P2 \trianglerighteq^{\flat} P1$
- No $\trianglerighteq^{\#/\flat}$ relation holds between P4 and others.

Inclusion in Strongly More/Less General Programs (not in the paper)

⌘ **Theorem**:

- If $P \trianglerighteq^{\#} Q$ then $A(P) \subseteq A(Q)$.
- If $P \trianglerighteq^{\flat} Q$ then $A(Q) \subseteq A(P)$.

⌘ The converse of each does not hold.

Generality in the Literature

- **Generality is discussed in ILP, but for the FO case only.**
- Sakama [IJCAI-2003; TCS 2005]
  - defines an ordering over extended logic programs based on multi-valued logics;
  - distinguishes definite and skeptical/credulous default information derived from a program.
  - Equivalent programs do not belong to the same equivalence class induced by Sakama's pre-order. (ex. $\{p\leftarrow\} \geq \{p\leftarrow \textbf{\textit{not}}\ q\}$)
- Zhang and Rounds [2001]
  - represent the semantics of programs using Smyth powerdomain;
  - do not consider comparison of multiple programs.
- Eiter, Tompits & Woltran [IJCAI-2005]
  - propose a general framework for comparing programs;
  - do not consider generality relations.

Conclusion

- ⌘ A formal theory for comparing generality between logic programs is proposed.
- ⌘ Both #- and ♭- generalities are defined in a way that weakly equivalent programs belong to the same equivalence class induced by these orderings.
- ⌘ Both minimal upper and maximal lower bonds can be defined for any pair of programs in these generality orderings.
- ⌘ #-general programs entails more skeptical consequences, while ♭- general programs entails more credulous consequences.
- ⌘ Both strong #- and strong ♭- generalities are defined in a way that strongly equivalent programs belong to the same equivalence class induced by these orderings.
- ⌘ The proposed orderings can be applied not only to ASP but also to any semantics based on minimal models.

Future Work

- Computing a more (or less) (strongly) general program for a given program
- Developing generalization/specialization methods in nonmonotonic ILP
- Investigating the notion of relative generality
- Relating strong generality to logic of here-and-there
- Extending generality orderings to non-minimal answer sets
- Exploring generality orderings in other nonmonotonic logics