Ordering Default Theories

Chiaki Sakama
Department of Computer and Communication Sciences
Wakayama University
Sakaedani, Wakayama 640 8510, Japan
sakama@sys.wakayama-u.ac.jp

Abstract

In first-order logic, a theory T_1 is considered stronger than another theory T_2 if every formula derived from T_2 is also derived from T_1. Such an order relation is useful to know relative value between different theories. In the context of default logic, a theory contains default information as well as definite information. To order default theories, it is necessary to assess the information content of a default theory. To this end, we introduce a multi-valued interpretation of default theories based on a nine-valued *bilattice*. It distinguishes definite and credulous/skeptical default information derived from a theory, and is used for ordering default theories based on their information contents. The technique is also applied to order nonmonotonic logic programs. The results of this paper provide a method for comparing different default theories and have important application to learning nonmonotonic theories.

1 Introduction

In knowledge representation based on logic, the relative value of a given theory is formally assessed by comparing the amount of information between theories. In first-order logic, a theory T_1 is considered *stronger* than another theory T_2 if every formula derived from T_2 is also derived from T_1 but not vice-versa (i.e., $T_1 \models T_2$ and $T_2 \not\models T_1$). For instance, the theory

$$T_1 = \{\, bird,\ \ bird \rightarrow flies \,\}$$

is stronger than the theory

$$T_2 = \{\, bird \vee flies \,\}.$$

In the context of *default logic* [Reiter, 1980], however, the problem is not so simple. For instance, consider the default theory:

$$T_3 = \{\, bird,\ \ \frac{bird : flies}{flies} \,\}.$$

In this case, both T_1 and T_3 imply $flies$, but the fact $flies$ from T_1 is a conclusion from definite information, while the same fact from T_3 is a conclusion from default information. Introducing another fact $\neg flies$ to each theory, the conclusion $flies$ is still derived from the inconsistent theory $T_1 \cup \{\, \neg flies \,\}$, while it is withdrawn from $T_3 \cup \{\, \neg flies \,\}$. Thus, two theories T_1 and T_3 have the same extension, but conclusions derived from T_1 are stronger than those of T_3.

To compare and order default theories, it is necessary to distinguish different sorts of information derived from a theory. Such consideration is meaningful and important with the following reasons.

- Studies in nonmonotonic logics have been centered on answering the question: "What information is concluded from a theory (with common-sense)?" On the other hand, few studies answer the question: "What sort of information is concluded from a theory?" Since default theories contain definite and default information, distinguishing different sorts of information is meaningful to assess the information content of a theory. Default theories contain *incomplete* information, so that the assessment provides a theoretical ground to measure the degree of "incompleteness" of a theory. These arguments are also effective in the field of *nonmonotonic logic programming* [Baral and Gelfond, 1994].
- It is important to know relative value between theories. A theory is considered more valuable than another theory if the former contains more information than the latter. Comparison of theories is especially important when there exist multiple sources of information as in *multi-agent systems*. In first-order logic, theories are ordered by logical entailment. In default logic, however, extensions of theories are not necessarily helpful for judging relative strength between theories (as presented above). To order default theories, it is necessary to provide a better ability of comparing default theories beyond their extensions. It should distinguish different sorts of information in a default theory, and order theories according to their information contents.
- In first-order logic, a theory is called *more general* than another theory if the former is stronger than the latter. Generality relations over first-order clauses have been extensively studied in the fields of *machine learning* and *inductive logic programming* [Nienhuys-Cheng and de Wolf, 1997]. In these fields, generalization is used as a basic operation for inductive learning, but it is unknown how to extend the notion to nonmonotonic the-

ories. To construct induction systems that learn nonmonotonic theories, it is necessary to extend the generalization operation and to build a theory for ordering nonmonotonic theories. Ordering default theories thus has potential application to the theory of induction in nonmonotonic logics and nonmonotonic inductive logic programming.

With these background and motivation, this paper studies a method for ordering default theories. To this end, we first provide a multi-valued interpretation for default theories based on a nine-valued *bilattice*. It can distinguish different sorts of information derived from default theories. We then introduce ordering over default theories, which orders different default theories based on the multi-valued interpretations of formulas. The techniques are also applied to order nonmonotonic logic programs under the *answer set semantics*. The rest of this paper is organized as follows. Section 2 reviews the framework of default logic. Section 3 develops a theory of ordering default theories. Section 4 applies the technique to nonmonotonic logic programming. Section 5 discusses related issues and Section 6 summarizes the paper.

2 Default Logic

A *default theory* is defined as a pair $\Delta = (D, W)$ where D is a set of default rules and W is a set of first-order formulas (called *facts*). A default rule (or simply *default*) is of the form:

$$\frac{\alpha : \beta_1, \ldots, \beta_n}{\gamma}$$

where $\alpha, \beta_1, \ldots, \beta_n$ and γ are first-order formulas and respectively called the *prerequisite*, the *justifications* and the *consequent*. In this paper, any default is assumed to have at least one justification ($n \geq 1$). A default theory is called *super-normal* if every default is of the form $\frac{:\gamma}{\gamma}$. As defaults and facts are syntactically distinguishable, we often write a default theory Δ as a set $W \cup D$ as far as no confusion arises. Any variable appearing in D and W is free and any default/fact with variables represents the set of its ground instances over the Herbrand universe of Δ. Throughout this paper we assume a default theory which is already ground-instantiated, i.e., for any default theory (D, W), D and W contain no variable. Also, a formula means a propositional formula unless stated otherwise.

A set E of formulas is an *extension* of (D, W) if it coincides with the smallest deductively closed set E' of formulas satisfying the conditions: (i) $W \subseteq E'$, and (ii) for any ground default $\alpha : \beta_1, \ldots, \beta_n / \gamma$ from D, $\alpha \in E'$ and $\neg\beta_i \notin E$ $(i = 1, \ldots, n)$ imply $\gamma \in E'$. A default theory may have none, one or multiple extensions in general. The set of all extensions of Δ is written as $\mathcal{EXT}(\Delta)$. Given a default theory Δ, a formula is a *credulous* conclusion of Δ if it belongs to some (but not all) extensions. By contrast, a formula is a *skeptical* conclusion of Δ if it belongs to all extensions. An extension E is *inconsistent* if it is the set of all formulas in the language.

Proposition 2.1 *[Reiter, 1980]* [1] *A default theory $\Delta = (D, W)$ has the inconsistent extension iff W is inconsistent.*

[1] This property holds for defaults with non-empty justifications.

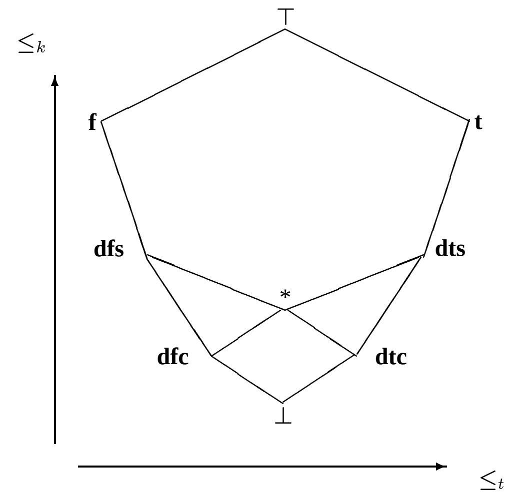

Figure 1: A lattice for logic IX

3 Ordering Default Theories

In classical logic, a formula F is interpreted true/false if $F/\neg F$ is a logical consequence of a theory; otherwise it is undefined. In default logic, on the other hand, a formula is either a definite consequence by W or a default consequence by D. Moreover, default consequences are brought by two different modes of inferences - skeptical or credulous reasoning. To characterize these different types of inferences, we first introduce a multi-valued logic for default reasoning.

Definition 3.1 The logic IX has the nine truth values $\mathbf{t}, \mathbf{f}, \top, \bot, \mathbf{dts}, \mathbf{dfs}, \mathbf{dtc}, \mathbf{dfc}, *$, which respectively mean *true*, *false*, *contradictory*, *undefined*, *skeptically true by default*, *skeptically false by default*, *credulously true by default*, *credulously false by default*, and *contradictory by default*.

The truth values of IX constitute a *bilattice* under the *knowledge ordering* $\leq_k$ and the *truth ordering* $\leq_t$ (Figure 1). Here, it holds that $\bot \leq_k \mathbf{dtc} \leq_k * \leq_k \mathbf{dts} \leq_k \mathbf{t} \leq_k \top$; $\bot \leq_k \mathbf{dfc} \leq_k * \leq_k \mathbf{dfs} \leq_k \mathbf{f} \leq_k \top$; $\mathbf{f} \leq_t \top \leq_t \mathbf{t}$; and $\mathbf{f} \leq_t \mathbf{dfs} \leq_t \mathbf{dfc} \leq_t \{*, \bot\} \leq_t \mathbf{dtc} \leq_t \mathbf{dts} \leq_t \mathbf{t}$.

In logic IX, negation $\neg$ is defined as: $\neg\top = \top$, $\neg\mathbf{t} = \mathbf{f}$, $\neg\mathbf{f} = \mathbf{t}$, $\neg\mathbf{dts} = \mathbf{dfs}$, $\neg\mathbf{dfs} = \mathbf{dts}$, $\neg\mathbf{dtc} = \mathbf{dfc}$, $\neg\mathbf{dfc} = \mathbf{dtc}$, $\neg* = *$, and $\neg\bot = \bot$. On the other hand, the disjunction $\vee$ and the conjunction $\wedge$ are respectively defined by the join operation and the meet operation with respect to the truth ordering in the bilattice. That is, $\mathbf{t} \vee x = \mathbf{t}$ for $x \in IX$, $* \vee \bot = \mathbf{dtc}$, $\top \wedge \mathbf{dfc} = \mathbf{f}$, and so on. It is easily verified that $\vee$ and $\wedge$ are associative, commutative, idempotent and absorptive.[2] Note that the truth-functional operations $\vee$ and $\wedge$ have their meaning supplied by the $\leq_t$ ordering, while we later order default theories by the $\leq_k$ ordering.

Under IX the interpretation of a formula in a default theory is defined as follows.

Definition 3.2 Given a default theory $\Delta = (D, W)$, the mapping ϕ_Δ associates a propositional formula F with a truth

[2] They are not distributive, e.g., $* \vee (\top \wedge \bot) \neq (* \vee \top) \wedge (* \vee \bot)$.

value of IX as follows:

$$\phi_\Delta(F) = \begin{cases} \mathbf{t} & \text{if } \mathcal{EXT}(\Delta) \neq \emptyset \text{ and } W \models F; \\ \mathbf{f} & \text{if } \mathcal{EXT}(\Delta) \neq \emptyset \text{ and } W \models \neg F; \\ \mathbf{dts} & \text{if } \mathcal{EXT}(\Delta) \neq \emptyset,\ W \not\models F \\ & \quad \text{and } \forall E \in \mathcal{EXT}(\Delta)\ \ F \in E; \\ \mathbf{dfs} & \text{if } \mathcal{EXT}(\Delta) \neq \emptyset,\ W \not\models \neg F \\ & \quad \text{and } \forall E \in \mathcal{EXT}(\Delta)\ \ \neg F \in E; \\ \mathbf{dtc} & \text{if } \exists E \in \mathcal{EXT}(\Delta) \text{ s.t. } F \in E \\ & \quad \text{and } \exists E' \in \mathcal{EXT}(\Delta) \text{ s.t. } F \notin E'; \\ \mathbf{dfc} & \text{if } \exists E \in \mathcal{EXT}(\Delta) \text{ s.t. } \neg F \in E \\ & \quad \text{and } \exists E' \in \mathcal{EXT}(\Delta) \text{ s.t. } \neg F \notin E'; \\ \bot & \text{otherwise.} \end{cases}$$

In particular, we write

$$\phi_\Delta(F) = \top \text{ if } \phi_\Delta(F) = \mathbf{t} \text{ and } \phi_\Delta(F) = \mathbf{f};$$
$$\phi_\Delta(F) = * \text{ if } \phi_\Delta(F) = \mathbf{dtc} \text{ and } \phi_\Delta(F) = \mathbf{dfc}.$$

Remark that formulas F and $\neg F$ are included in every extension of Δ iff W is inconsistent (Proposition 2.1). So it does not happen that both $\phi_\Delta(F) = \mathbf{dts}$ and $\phi_\Delta(F) = \mathbf{dfs}$ for any F.

The mapping ϕ_Δ provides multi-valued interpretations of formulas in a default theory. Intuitively, $\phi_\Delta(F) \in \{\mathbf{t}, \mathbf{f}\}$ means that F or $\neg F$ is a definite conclusion from W. When $\phi_\Delta(F) = \top$ for some formula F, $\phi_\Delta(G) = \top$ for any formula G. This is because in this case W is inconsistent and entails every formula.[3] On the other hand, when $\phi_\Delta(F) \in \{\mathbf{dts}, \mathbf{dfs}\}$ (resp. $\phi_\Delta(F) \in \{\mathbf{dtc}, \mathbf{dfc}\}$), F or $\neg F$ is a default conclusion inferred skeptically (resp. credulously) from Δ. When $\phi_\Delta(F) = *$, a formula F belongs to some extension and its negation $\neg F$ belongs to another extension. In this case, the truth value of F is contradictory by default. ϕ_Δ maps a formula into $\bot$ if it is included in no extension. In particular, every formula has the value $\bot$ if Δ has no extension.

Example 3.1 Let Δ be the theory:

$$bird, \quad \frac{bird : flies}{flies},$$

which has the single extension $Th(\{\, bird,\ flies \,\})$. Then $\phi_\Delta(bird) = \mathbf{t}$, $\phi_\Delta(flies) = \mathbf{dts}$, and $\phi_\Delta(bird \to flies) = \mathbf{dts}$, etc.

Example 3.2 Let Δ be the theory:

$$\frac{: \neg rh\text{-}broken \wedge lh\text{-}broken}{lh\text{-}broken}, \quad \frac{: \neg lh\text{-}broken \wedge rh\text{-}broken}{rh\text{-}broken},$$

which has two extensions $Th(\{lh\text{-}broken\})$ and $Th(\{rh\text{-}broken\})$. Then $\phi_\Delta(lh\text{-}broken) = \phi_\Delta(rh\text{-}broken) = \mathbf{dtc}$, $\phi_\Delta(lh\text{-}broken \vee rh\text{-}broken) = \mathbf{dts}$ and $\phi_\Delta(lh\text{-}broken \wedge rh\text{-}broken) = \bot$.

Example 3.3 Let Δ be the theory:

$$quaker \wedge republican,$$
$$\frac{quaker : pacifist}{pacifist}, \quad \frac{republican : \neg pacifist}{\neg pacifist},$$

which has two extensions $Th(\{quaker \wedge republican,\ pacifist\})$ and $Th(\{quaker \wedge republican, \neg pacifist\})$. Then $\phi_\Delta(quaker \wedge republican) = \mathbf{t}$ and $\phi_\Delta(pacifist) = *$.

[3] In this sense, our logic is not "paraconsistent".

The followings are some properties of ϕ_Δ.

Proposition 3.1 *Given two formulas F and G, $F \equiv G$ implies $\phi_\Delta(F) = \phi_\Delta(G)$.*

Proposition 3.2 *For formulas F and G,*

- $\phi_\Delta(\neg F) = \neg\phi_\Delta(F)$.
- $\phi_\Delta(F) \geq_k \phi_\Delta(G)$ *iff* $\phi_\Delta(\neg F) \geq_k \phi_\Delta(\neg G)$.
- $\phi_\Delta(F) \geq_t \phi_\Delta(G)$ *iff* $\phi_\Delta(\neg G) \geq_t \phi_\Delta(\neg F)$.
- $\neg(\phi_\Delta(F) \vee \phi_\Delta(G)) = \neg\phi_\Delta(F) \wedge \neg\phi_\Delta(G)$.
- $\neg(\phi_\Delta(F) \wedge \phi_\Delta(G)) = \neg\phi_\Delta(F) \vee \neg\phi_\Delta(G)$.

Proposition 3.3 *For formulas F and G,*

- $\phi_\Delta(F \vee G) \geq_t \phi_\Delta(F) \vee \phi_\Delta(G)$.
- $\phi_\Delta(F \wedge G) \leq_t \phi_\Delta(F) \wedge \phi_\Delta(G)$.
- $\phi_\Delta(F \to G) \geq_t \phi_\Delta(\neg F) \vee \phi_\Delta(G)$.

Proof: By $\phi_\Delta(F \vee G) \geq_t \phi_\Delta(F)$ and $\phi_\Delta(F \vee G) \geq_t \phi_\Delta(G)$, $\phi_\Delta(F \vee G) \geq_t \phi_\Delta(F) \vee \phi_\Delta(G)$ holds. Also, $\phi_\Delta(F \wedge G) \leq_t \phi_\Delta(F)$ and $\phi_\Delta(F \wedge G) \leq_t \phi_\Delta(G)$ imply $\phi_\Delta(F \wedge G) \leq_t \phi_\Delta(F) \wedge \phi_\Delta(G)$. The result for implication follows by the relation $\phi_\Delta(F \to G) = \phi_\Delta(\neg F \vee G)$. □

Example 3.4 In Example 3.2, $\phi_\Delta(lh\text{-}broken \vee rh\text{-}broken) \geq_t \phi_\Delta(lh\text{-}broken) \vee \phi_\Delta(rh\text{-}broken)$ and $\phi_\Delta(lh\text{-}broken \wedge rh\text{-}broken) \leq_t \phi_\Delta(lh\text{-}broken) \wedge \phi_\Delta(rh\text{-}broken)$.

Thus, the degree of truth of an internal disjunction (resp. conjunction) is generally higher (resp. lower) than that of an external one. By contrast, there are no general relations between $\phi_\Delta(F \vee G)$ and $\phi_\Delta(F) \vee \phi_\Delta(G)$; and $\phi_\Delta(F \wedge G)$ and $\phi_\Delta(F) \wedge \phi_\Delta(G)$ under the $\leq_k$ ordering.

Next we introduce ordering between default theories.

Definition 3.3 Let Δ_1 and Δ_2 be two default theories which have the same underlying language. Then, Δ_1 is *stronger* than Δ_2 (written as $\Delta_2 \leq_{DL} \Delta_1$) if $\phi_{\Delta_2}(F) \leq_k \phi_{\Delta_1}(F)$ for any formula F in the language. We write $\Delta_1 \simeq_{DL} \Delta_2$ if $\Delta_1 \leq_{DL} \Delta_2$ and $\Delta_2 \leq_{DL} \Delta_1$.

The relation $\leq_{DL}$ is a pre-order, i.e., a reflexive and transitive relation on the set of all default theories in the language. Throughout the paper, when we compare different default theories, we assume that they have the same underlying language.

Intuitively, a default theory Δ_1 is stronger than another default theory Δ_2 if Δ_1 entails more certain information than Δ_2. In other words, when $\Delta_2 \leq_{DL} \Delta_1$, conclusions derived from Δ_1 are relatively more stable and reliable than those derived from Δ_2. The 'stronger' relation reduces to the relation between (propositional) first-order theories when default theories have no defaults.

Proposition 3.4 *Let $\Delta_1 = (\emptyset, W_1)$ and $\Delta_2 = (\emptyset, W_2)$ be two default theories. Then, $\Delta_2 \leq_{DL} \Delta_1$ iff $W_1 \models W_2$.*

Thus, the relation $\leq_{DL}$ is a natural extension of the one for (propositional) first-order theories.

Proposition 3.5 *Given two default theories Δ_1 and Δ_2, $\Delta_1 \simeq_{DL} \Delta_2$ implies $\mathcal{EXT}(\Delta_1) = \mathcal{EXT}(\Delta_2)$, but not vice-versa.*

The above proposition presents that the order-equivalence relation $\simeq_{DL}$ provides an equivalence relation which is stronger than the equivalence based on extensions.

Example 3.5 (introductory example) Let Δ_1 and Δ_2 be two default theories:

$$\Delta_1: \quad bird, \quad bird \rightarrow flies,$$
$$\Delta_2: \quad bird, \quad \frac{bird : flies}{flies},$$

where $\phi_{\Delta_1}(bird) = \phi_{\Delta_2}(bird) = \mathbf{t}$, $\phi_{\Delta_1}(flies) = \mathbf{t}$, $\phi_{\Delta_2}(flies) = \mathbf{dts}$. Then, $\Delta_2 \leq_{DL} \Delta_1$.

On the other hand, given

$$\Delta_3: \ bird, \quad \frac{: bird \rightarrow flies}{bird \rightarrow flies},$$

$\Delta_2 \simeq_{DL} \Delta_3$. Note that all Δ_1, Δ_2 and Δ_3 have the same extension $Th(\{\, bird,\ flies \,\})$.

The order $\leq_{DL}$ is nonmonotonic with respect to the increase of information.

Proposition 3.6 *Let Δ_1 and Δ_2 be two default theories and F a formula. Then, $\Delta_1 \leq_{DL} \Delta_2$ implies neither $\Delta_1 \leq_{DL} \Delta_2 \cup \{F\}$ nor $\Delta_1 \cup \{F\} \leq_{DL} \Delta_2 \cup \{F\}$. In particular, $\Delta_1 \not\leq_{DL} \Delta_1 \cup \{F\}$ in general.*

Example 3.6 Let Δ_1 and Δ_2 be two default theories:

$$\Delta_1: \quad \frac{: p \wedge \neg q}{p},$$
$$\Delta_2: \quad \frac{: p \wedge \neg q}{p}, \quad r,$$

where $\phi_{\Delta_1}(p) = \phi_{\Delta_2}(p) = \mathbf{dts}$, $\phi_{\Delta_1}(q) = \phi_{\Delta_2}(q) = \bot$, $\phi_{\Delta_1}(r) = \bot$ and $\phi_{\Delta_2}(r) = \mathbf{t}$. Then, $\Delta_1 \leq_{DL} \Delta_2$ holds. Let $F = (r \rightarrow q)$. Then, $\Delta_1 \not\leq_{DL} \Delta_2 \cup \{F\}$ and $\Delta_1 \cup \{F\} \not\leq_{DL} \Delta_2 \cup \{F\}$ by $\phi_{\Delta_1 \cup \{F\}}(p) = \mathbf{dts}$ and $\phi_{\Delta_2 \cup \{F\}}(p) = \bot$.

The introduction of new information may block the application of some default rules, which would cause the withdrawal of some default conclusions in a theory. This is a typical feature of default reasoning.

We finally provide a connection between the order relation $\leq_{DL}$ and default extensions.

Theorem 3.7 *Let $\Delta_1 = (D_1, W_1)$ and $\Delta_2 = (D_2, W_2)$ be two default theories. Then, $\Delta_1 \leq_{DL} \Delta_2$ if the following conditions are satisfied:*

1. $W_2 \models W_1$,
2. $\forall E_2 \in \mathcal{EXT}(\Delta_2)\ \exists E_1 \in \mathcal{EXT}(\Delta_1)$ *s.t.* $E_1 \subseteq E_2$,
3. $\forall E_1 \in \mathcal{EXT}(\Delta_1)\ \exists E_2 \in \mathcal{EXT}(\Delta_2)$ *s.t.* $E_1 \subseteq E_2$.

Proof: Let F be a formula. By $W_2 \models W_1$, $\mathbf{t} \leq_k \phi_{\Delta_1}(F)$ implies $\mathbf{t} \leq_k \phi_{\Delta_2}(F)$, and $\mathbf{f} \leq_k \phi_{\Delta_1}(F)$ implies $\mathbf{f} \leq_k \phi_{\Delta_2}(F)$. Suppose $\phi_{\Delta_1}(F) = \mathbf{dts}$. In this case, $\mathcal{EXT}(\Delta_1) \neq \emptyset$, $W \not\models F$ and F is included in every extension of Δ_1. By the second condition, F is included in every extension of Δ_2. Then, $\mathbf{dts} \leq_k \phi_{\Delta_2}(F)$. Similarly, it is shown that $\phi_{\Delta_1}(F) = \mathbf{dfs}$ implies $\mathbf{dfs} \leq_k \phi_{\Delta_2}(F)$. Next, suppose $\phi_{\Delta_1}(F) = \mathbf{dtc}$. In this case, F is included in some (but not every) extension of Δ_1. By the third condition, F is included in some extension of Δ_2. Then, $\mathbf{dtc} \leq_k \phi_{\Delta_2}(F)$. Similarly, it is shown that $\phi_{\Delta_1}(F) = \mathbf{dfc}$ implies $\mathbf{dfc} \leq_k \phi_{\Delta_2}(F)$. Finally, suppose $\phi_{\Delta_1}(F) = \bot$. Since $\phi_{\Delta_2}(F) = x$ with $\bot \leq_k x$, $\bot \leq_k \phi_{\Delta_2}(F)$ holds. □

Theorem 3.7 provides a sufficient condition to judge $\Delta_1 \leq_{DL} \Delta_2$ using extensions of default theories.

4 Ordering Nonmonotonic Logic Programs

In logic programming, default reasoning is realized by *negation as failure* (NAF). Logic programs containing NAF are called *nonmonotonic logic programs*.

Nonmonotonic logic programs considered in this paper are the class of *extended logic programs* (ELPs) [Gelfond and Lifschitz, 1991], which contain two kinds of negation; explicit (or classical) negation $\neg$ and NAF (or default negation) not. An extended logic program (or simply a program) is a set of *rules* of the form:

$$L_0 \leftarrow L_1, \ldots, L_m, not\, L_{m+1}, \ldots, not\, L_n \quad (n \geq m)$$

where each L_i $(0 \leq i \leq n)$ is a positive/negative literal and not represents NAF. The literal L_0 is the *head* and the conjunction $L_1, \ldots, L_m, not\, L_{m+1}, \ldots, not\, L_n$ is the *body* of the rule. A rule or a program is called *not-free* if it contains no NAF (i.e., $m = n$). Given an ELP Π, the set of not-free rules from Π is denoted by Π^+. A rule with the empty body $L \leftarrow$ is identified with the literal L. The head of any rule is non-empty.[4] Like default theories, every variable in a program is interpreted as a free variable. A program Π is semantically identified with its ground instantiation, i.e., the set of ground rules obtained from Π by substituting variables with elements of the Herbrand universe of Π in every possible way. We handle ground programs throughout the paper.

The semantics of ELPs is given by the *answer set semantics* [Gelfond and Lifschitz, 1991]. Let Lit be the set of all ground literals in the language of a program (called the *literal base*). Suppose an ELP Π and a set of literals $S (\subseteq Lit)$. Then, the *reduct* Π^S is the program which contains the ground rule $L_0 \leftarrow L_1, \ldots, L_m$ iff there is a rule $L_0 \leftarrow L_1, \ldots, L_m, not\, L_{m+1}, \ldots, not\, L_n$ in the ground instantiation of Π such that $\{\, L_{m+1}, \ldots, L_n \,\} \cap S = \emptyset$. Given a not-free ELP Π, $Cn(\Pi)$ denotes the smallest set of ground literals which is (i) *closed* under Π, i.e., for every ground rule $L_0 \leftarrow L_1, \ldots, L_m$ from the ground instantiation of Π, $\{L_1, \ldots, L_m\} \subseteq Cn(\Pi)$ implies $L_0 \in Cn(\Pi)$; and (ii) *logically closed*, i.e., it is either consistent or equal to Lit. Given an ELP Π and a set S of literals, S is an *answer set* of Π if $S = Cn(\Pi^S)$.

Answer sets represent possible beliefs of a program, and an ELP may have none, one, or multiple answer sets. In particular, every not-free ELP has the unique answer set. An answer set is *consistent* if it is not Lit. The set of all answer sets of an ELP Π is written as $\mathcal{AS}(\Pi)$.

Proposition 4.1 *An ELP Π has the unique answer set Lit iff $Cn(\Pi^+) = Lit$.*

[4] Under the answer set semantics which we consider in this paper, a rule with the empty head $\leftarrow F$ is expressed by the semantically equivalent rule $L \leftarrow F,\ not\, L$ with a literal L.

According to [Gelfond and Lifschitz, 1991], the rule $L_0 \leftarrow L_1, \ldots, L_m, \mathit{not}\, L_{m+1}, \ldots, \mathit{not}\, L_n$ is interpreted as the default rule:

$$\frac{L_1 \wedge \cdots \wedge L_m : \neg L_{m+1}, \ldots, \neg L_n}{L_0}$$

where $\neg\neg L = L$ for a positive literal L. In this case, there is a 1-1 correspondence between the answer sets of a program and the extensions of the corresponding default theory. [5]

Proposition 4.2 *[Gelfond and Lifschitz, 1991] Let Π be an ELP and Δ_Π its corresponding default theory. If S is an answer set of Π, then the deductive closure of S is an extension of Δ_Π. Conversely, every extension of Δ_Π is the deductive closure of exactly one answer set of Π.*

Using the correspondence, a multi-valued interpretation for ELPs is defined under the logic IX.

Definition 4.1 Given an ELP Π, the mapping ϕ_Π associates a positive literal $L \in Lit$ with a truth value of IX as follows:

$$\phi_\Pi(L) = \begin{cases} \mathbf{t} & \text{if } \mathcal{AS}(\Pi) \neq \emptyset \text{ and } L \in Cn(\Pi^+)\,; \\ \mathbf{f} & \text{if } \mathcal{AS}(\Pi) \neq \emptyset \text{ and } \neg L \in Cn(\Pi^+)\,; \\ \mathbf{dts} & \text{if } \mathcal{AS}(\Pi) \neq \emptyset,\ L \notin Cn(\Pi^+) \\ & \text{and } \forall S \in \mathcal{AS}(\Pi)\ \ L \in S\,; \\ \mathbf{dfs} & \text{if } \mathcal{AS}(\Pi) \neq \emptyset,\ \neg L \notin Cn(\Pi^+) \\ & \text{and } \forall S \in \mathcal{AS}(\Pi)\ \ \neg L \in S\,; \\ \mathbf{dtc} & \text{if } \exists S \in \mathcal{AS}(\Pi) \text{ s.t. } L \in S \\ & \text{and } \exists T \in \mathcal{AS}(\Pi) \text{ s.t. } L \notin T\,; \\ \mathbf{dfc} & \text{if } \exists S \in \mathcal{AS}(\Pi) \text{ s.t. } \neg L \in S \\ & \text{and } \exists T \in \mathcal{AS}(\Pi) \text{ s.t. } \neg L \notin T\,; \\ \bot & \text{otherwise.} \end{cases}$$

In particular, we write

$$\phi_\Pi(L) = \top \text{ if } \phi_\Pi(L) = \mathbf{t} \text{ and } \phi_\Pi(L) = \mathbf{f};$$
$$\phi_\Pi(L) = * \text{ if } \phi_\Pi(L) = \mathbf{dtc} \text{ and } \phi_\Pi(L) = \mathbf{dfc}.$$

Remark that literals L and $\neg L$ are included in every answer set of Π iff they are in $Cn(\Pi^+)$ (Proposition 4.1). So it does not happen that $\phi_\Pi(L)$ takes both $\mathbf{dts}$ and $\mathbf{dfs}$ for any L.

The intuitive meaning of ϕ_Π is analogous to that of ϕ_Δ.

Example 4.1 Let Π be the program:

$$p \leftarrow \mathit{not}\, q, \quad q \leftarrow \mathit{not}\, p, \quad r \leftarrow \mathit{not}\, \neg s,$$

which has two answer sets $\{p, r\}$ and $\{q, r\}$. Then $\phi_\Pi(p) = \phi_\Pi(q) = \mathbf{dtc}$, $\phi_\Pi(r) = \mathbf{dts}$, and $\phi_\Pi(s) = \bot$.

Example 4.2 Let Π be the program:

$$p \leftarrow \mathit{not}\, \neg p, \quad \neg p \leftarrow \mathit{not}\, p,$$

which has two answer sets $\{p\}$ and $\{\neg p\}$. Then, $\phi_\Pi(p) = *$.

ϕ_Π has properties obtained from Proposition 3.2 by replacing ϕ_Δ with ϕ_Π and formulas with literals.

Ordering between ELPs is defined as follows.

[5] Precisely speaking, *not*-free rules in an ELP correspond to justification-free defaults. Although we supposed defaults with non-empty justifications in the previous sections, the following discussion is valid apart from the results of Section 3.

Definition 4.2 Let Π_1 and Π_2 be two ELPs which have the same literal base Lit. Then, Π_1 is *stronger* than Π_2 under the answer set semantics (written as $\Pi_2 \leq_{AS} \Pi_1$) if $\phi_{\Pi_2}(L) \leq_k \phi_{\Pi_1}(L)$ for any literal $L \in Lit$. We write $\Pi_1 \simeq_{AS} \Pi_2$ if $\Pi_1 \leq_{AS} \Pi_2$ and $\Pi_2 \leq_{AS} \Pi_1$.

The relation $\leq_{AS}$ is a pre-order on the set of all ELPs in the language. A program Π_1 is stronger than Π_2 if the answer sets of Π_1 entail more certain information than Π_2. Different from the case of default logic, we compare programs in terms of literals included in answer sets. This is because in nonmonotonic logic programs the meaning of a program is determined not by individual rules in a program, but by consequent literals included in selected models of a program. Thus, we capture the information content of a program as consequences brought by answer sets.

Proposition 4.3 *Let Π_1 and Π_2 be two not-free ELPs which have the same literal base. Then, $\Pi_1 \leq_{AS} \Pi_2$ iff $Cn(\Pi_1) \subseteq Cn(\Pi_2)$.*

Proposition 4.4 *Given two ELPs Π_1 and Π_2, $\Pi_1 \simeq_{AS} \Pi_2$ implies $\mathcal{AS}(\Pi_1) = \mathcal{AS}(\Pi_2)$, but not vice-versa.*

Example 4.3 Let Π_1 and Π_2 be two programs:

$$\Pi_1: \quad p \leftarrow \neg q, \quad \neg q \leftarrow,$$
$$\Pi_2: \quad p \leftarrow \mathit{not}\, q, \quad \neg q \leftarrow \mathit{not}\, q,$$

where Π_1 and Π_2 have the same answer set $\{p, \neg q\}$. Then, $\phi_{\Pi_1}(p) = \mathbf{t}$, $\phi_{\Pi_1}(q) = \mathbf{f}$, $\phi_{\Pi_2}(p) = \mathbf{dts}$, and $\phi_{\Pi_2}(q) = \mathbf{dfs}$. Thus, $\Pi_2 \leq_{AS} \Pi_1$.

It is easily verified that the order $\leq_{AS}$ has nonmonotonic properties which correspond to Proposition 3.6 with respect to the introduction of new rules to a program.

A connection between the order relation $\leq_{AS}$ and answer sets is given as follows (The proof is similar to Theorem 3.7).

Theorem 4.5 *Let Π_1 and Π_2 be two ELPs which have the same literal base. Then, $\Pi_1 \leq_{AS} \Pi_2$ if the following conditions are satisfied:*

1. $Cn(\Pi_1^+) \subseteq Cn(\Pi_2^+)$,
2. $\forall S \in \mathcal{AS}(\Pi_2)\ \exists T \in \mathcal{AS}(\Pi_1)$ *s.t.* $T \subseteq S$,
3. $\forall T \in \mathcal{AS}(\Pi_1)\ \exists S \in \mathcal{AS}(\Pi_2)$ *s.t.* $T \subseteq S$.

5 Discussion

In the context of multi-valued logics, Ginsberg [1988] firstly introduces a bilattice for default logic. He distinguishes definite and default conclusions obtained from a (super-normal) default theory using the bilattice of Figure 2. However, Ginsberg's bilattice is seven-valued and does not distinguish between skeptical and credulous default conclusions. For instance, suppose the super-normal default theory $\Delta = \{\frac{:p \wedge q}{p \wedge q}, \frac{:\neg p}{\neg p}\}$ which has two default extensions $Th(\{p \wedge q\})$ and $Th(\{\neg p\})$. Then, $\phi_\Delta(p) = *$, $\phi_\Delta(q) = \mathbf{dtc}$, and $\phi_\Delta(\neg p \vee q) = \mathbf{dts}$ in our framework, while Ginsberg interprets p as $*$ but handles both q and $\neg p \vee q$ as $\mathbf{dt}$. Thus, to distinguish skeptical/credulous default inference, additional truth values are necessary as introduced in this paper. Dionisio *et al.* [1998] distinguish skeptical/credulous default inference in super-normal default theories using modal logic.

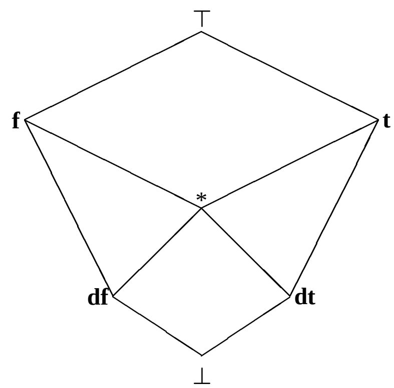

Figure 2: Ginsberg's bilattice for default logic

Their goal is reasoning about defaults and is not ordering default theories.

In logic programming, Fitting [1991] uses bilattices to characterize the semantics of normal logic programs. He uses a four-valued logic and does not distinguish definite and default information. Dix [1992] uses the knowledge ordering under a three-valued logic to compare information obtained from a single normal logic program under different semantics. This is in contrast to our approach in which we compare different programs under the single answer set semantics. Lattice-valued logics are also used for characterizing the "paraconsistent" semantics of logic programs [Damásio and Pereira, 1998]. To our best knowledge, however, the 9-valued bilattice used in this paper never appears in the literature. Moreover, existing studies all use multi-valued logics to provide a semantics of a single program, while we use them to compare information between different programs. The order-equivalence provides a stronger relation than the usual model-based equivalence (Proposition 4.4). On the other hand, it is known another *strong equivalence* relation between logic programs [Lifschitz $et\,al.$, 2001]. At the moment, we have an evidence that there is no stronger/weaker relation between the strong equivalence and the order-equivalence in general.

From the computational viewpoint, there is a difficulty for directly computing ϕ_Δ for an arbitrary formula F. This is due to the fact that the interpretation ϕ_Δ of a formula F is generally not constructive by those of the sub-formulas of F (Proposition 3.3). The same problem happens in the restricted class of super-normal default theories [Ginsberg, 1988]. For checking an order between default theories, however, Theorem 3.7 provides a sufficient condition to judge the relation $\Delta_1 \leq_{DL} \Delta_2$ using default extensions. In the context of logic programming, $\Pi_1 \leq_{AS} \Pi_2$ is checked by Theorem 4.5 using the existing procedures for computing answer sets.

In the fields of machine learning and inductive logic programming, a theory of generalization has been extensively studied in the context of first-order logic [Nienhuys-Cheng and de Wolf, 1997]. However, generalization under logical entailment $\models$ is not directly applicable to default theories and nonmonotonic logic programs. A default ordering introduced in this paper can order default theories and nonmonotonic logic programs, thereby could give a theoretical ground for inductive generalization in nonmonotonic logic programs. For instance, for the programs $\Pi = \{\, flies(x) \leftarrow bird(x),\quad bird(tweety) \leftarrow \}$ and $\Pi' = \{\, flies(x) \leftarrow bird(x), not\ abnormal(x),\quad bird(tweety) \leftarrow \}$, the relation $\Pi' \leq_{AS} \Pi$ holds ($flies(tweety)$ has the value $\mathbf{t}$ in Π, while it has the value $\mathbf{dts}$ in Π'). Thus, if we read the order $\leq_{AS}$ as "more general", Π is considered a generalization of Π'. This coincides with the view in the ILP literature [Bain and Muggleton, 1992] in which Π' is a specialization of Π.

6 Conclusion

In this paper, we introduced a multi-valued interpretation of default theories, which can distinguish definite and skeptical/credulous default consequences. Based on this, we developed a theory for ordering default theories and applied the technique to ordering nonmonotonic logic programs. The 9-valued bilattice is used for characterizing other nonmonotonic formalisms which have the same inference modes as default logic. The results of this paper provide a method of comparing default theories or nonmonotonic logic programs in a manner different from the conventional extension-based or model-based standpoint. The proposed framework is considered to have potential application to inductive learning in nonmonotonic logics, which we will investigate in future study.

References

[Bain and Muggleton, 1992] M. Bain and S. Muggleton. Non-monotonic learning. In: Muggleton (ed.), *Inductive Logic Programming*, Academic Press, pp. 145–161, 1992.

[Baral and Gelfond, 1994] C. Baral and M. Gelfond. Logic programming and knowledge representation. *J. Logic Programming* 19/20:73–148, 1994.

[Damásio and Pereira, 1998] C. V. Damásio and L. M. Pereira. A survey of paraconsistent semantics for logic programs. In: *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 2, D. M. Gabbay and Ph. Smets (eds.), Kluwer Academic, pp. 241–320, 1998.

[Dionísio $et\,al.$, 1998] F. M. Dionísio, S. Brass, M. Ryan, U. Lipeck. Hypothetical reasoning with defaults. In: *Proc. Workshop on Computational Aspects of NMR*, 1998.

[Dix, 1992] J. Dix. A framework for representing and characterizing semantics of logic programs. In: *Proc. KR'92*, Morgan Kaufmann, pp. 591–602, 1992.

[Fitting, 1991] M. Fitting. Bilattices and the semantics of logic programming. *J. Logic Program.* 11:91–116, 1991.

[Gelfond and Lifschitz, 1991] M. Gelfond and V. Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385, 1991.

[Ginsberg, 1988] M. L. Ginsberg. Multi-valued logics: a uniform approach to inference in artificial intelligence. *Computational Intelligence* 4:265–316, 1988.

[Lifschitz $et\,al.$, 2001] V. Lifschitz, D. Pearce, and A. Valverde. Strongly equivalent logic programs. *ACM Trans. Computational Logic* 2:526–541, 2001.

[Nienhuys-Cheng and de Wolf, 1997] S.-H. Nienhuys-Cheng and R. de Wolf. *Foundations of inductive logic programming. LNAI* 1228, Springer, 1997.

[Reiter, 1980] R. Reiter. A logic for default reasoning. *Artificial Intelligence* 13:81–132, 1980.