Dualities between Alternative Semantics for Logic Programming and Nonmonotonic Reasoning

Chitta R. Baral and V.S. Subrahmanian*
Department of Computer Science
University of Maryland
College Park, MD 20742
U.S.A.

September 21, 1998

Abstract

The Gelfond-Lifschitz operator [GL88] associated with a logic program (and likewise the operator associated with default theories by Reiter) exhibits oscillating behavior. In the case of logic programs, there is always at least one finite, non-empty collection of Herbrand interpretations around which the Gelfond-Lifschitz [GL88] operator "bounces around". The same phenomenon occurs with default logic when Reiter's operator Γ_Δ is considered. Based on this, a "stable class" semantics and "extension class" semantics was proposed in [BS90]. The main advantage of this semantics was that it was defined for all logic programs (and default theories), and that this definition was modelled using the standard operators existing in the literature such as Reiter's Γ_Δ operator. In this paper, our primary aim is to prove that there is a very interesting *duality* between stable class theory and the well founded semantics for logic programming. In the stable class semantics, classes that were minimal with respect to Smyth's power-domain ordering were selected. We show that the well founded semantics precisely corresponds to a class that is minimal w.r.t. Hoare's power domain ordering: the well known dual of Smyth's ordering. Besides this elegant duality, this immediately suggests how to define a well-founded semantics for default logic in such a way that the dualities that hold for logic programming continue to hold for default theories. We show how the same technique may be applied to "strong" auto-epistemic logic: the logic of strong expansions proposed by Marek and Truszczynski [MT89a].

1 Introduction

Various alternative semantics for logic programming with non-monotonic modes of negation have been proposed. Of these, the most popular semantics thus far are the well-founded semantics [VRS88] and the stable model semantics [GL88]. In this paper, we study the operator (we call it F_P) mapping interpretations to interpretations defined by Gelfond and

*Joint appointment with the University of Maryland Institute for Advanced Computer Studies.

Lifschitz[GL88]. Fixed-points of F_P are the stable models of P. It has been shown by Baral and Subrahmanian [BS90] that F_P always "oscillates" amongst a collection of interpretations, i.e. for every program P, there is at least one non-empty set, $\mathcal{A}$, of interpretations such that $\mathcal{A} = \{F_P(I) \mid I \in \mathcal{A}\}$. Such sets $\mathcal{A}$ are called *stable classes* and a stable class semantics was defined in [BS90].

In this paper, we prove that the well-founded semantics for logic programs, and the stable class semantics both select certain minimal (w.r.t. certain orderings) stable classes. We show here that under two well-known orderings on power domains (Hoare and Smyth orderings [Smy78]) the well-founded semantics corresponds *precisely* to stable classes that are minimal w.r.t. Hoare's ordering. We show that this phenomenon also occurs in default logic and for suitable structures in auto-epistemic logic. As a consequence, we are able to present a well-founded semantics for default and auto-epistemic logics. This provides an iterative fixpoint semantics for default theories and auto-epistemic theories. Furthermore, the Hoare-Smyth duality that occurs for logic programming, also holds in the case of default and auto-epistemic logic. An additional advantage is that we may, based on our intended applications, choose different orderings (e.g. orderings based on cardinality) on the set of stable classes.

Przymusinski [Prz89] has defined a well-founded semantics for default logics and auto-epistemic logics. However, these semantics are based on the construction of new three valued reformulations of default and auto-epistemic logic. We work only with the usual two-valued formalizations of default and auto-epistemic logic.

2 Duality in the semantics of logic programs

2.1 Preliminaries

We assume that all logic programs are (possibly infinite) sets of ground clauses. A clause is a sentence of the form

$$A \leftarrow B_1 \,\&\, \ldots \,\&\, B_n \,\&\, \neg D_1 \,\&\, \cdots \,\&\, \neg D_m$$

where $A, B_1, \ldots, B_n, D_1, \ldots, D_m$ are all atoms and $n, m \geq 0$. Note that there is no loss of generality in writing negated literals to the right of positive literals.

The Herbrand Base of a logic program P is denoted by B_P. An Herbrand interpretation I is any subset of the Herbrand Base. We use the notation $\overline{I}$ to denote the complement of I, i.e. $\overline{I} = (B_P - I)$. If $I \subseteq B_P$ is any interpretation, then $\neg I$ denotes the set $\{\neg A \mid A \in I\}$. If $I \subseteq B_P$ is any interpretation, then $\tilde{I}$ denotes the set $\{\tilde{p}(\vec{t}) \mid p(\vec{t}) \in I\}$. Thus, $\tilde{I}$ is a set of atoms obtained by renaming all predicate symbols p in I by $\tilde{p}$.

Suppose P is a logic program and $I \subseteq B_P$. The *Gelfond-Lifschitz* transformation of P, denoted P^I, is the logic program defined as follows:

The clause $A \leftarrow B_1 \,\&\ldots\&\, B_n$, $n \geq 0$, is in P^I iff there exists a clause

$$A \leftarrow B_1 \,\&\ldots\&\, B_n \,\&\, \neg D_1 \,\&\cdots\&\, \neg D_m$$

$(m \geq 0)$ in P such that $I \cap \{D_1, \ldots, D_m\} = \emptyset$. Nothing else is in P^I. Thus, P^I is a negation-free logic program.

Given a program P and an Herbrand interpretation I, we may define an operator, F_P, associated with P as follows: $F_P(I) = T_{P^I} \uparrow \omega$, i.e. $F_P(I)$ is the least Herbrand model of the negation free logic program P^I. We assume that readers are familiar with the operator T_P associated with the logic program P. (cf. [L87]).

Definition 2.1 (Gelfond and Lifschitz) I is a *stable* model of P iff $I = F_P(I)$.

Proposition 2.1 (Baral and Subrahmanian) Let P be any logic program. Then: F_P is anti-monotone, i.e. if $I_1 \subseteq I_2$, then $F_P(I_2) \subseteq F_P(I_1)$. □

Since, F_P is anti-monotonic, F_P^2, the function that applies F_P twice is monotonic.

Corollary 2.1 F_P^2 is monotonic. □

Definition 2.2 Let $S = \{I_i \mid i \in \mathcal{A}\}$ be a *finite* set of interpretations. S is said to be a *stable class* of program P iff $S = \{F_P(I_i) \mid i \in \mathcal{A}\}$.

Intuitively strict stable classes correspond to a <u>finite</u> set of interpretations amongst which F_P "oscillates". Basically, given a logic program P, we can associate a directed graph with P. The nodes of the graph are Herbrand interpretations. There is an arc from interpretation I to interpretation J just in case $J = F_P(I)$. The cycles in this graph correspond to strict stable classes. [BS90] contains a detailed description of the properties of these graphs.

Definition 2.3 S is said to be a *strict stable class* of a logic program P iff no proper non-empty subset of S is a stable class of P.

Example 2.1 Consider the logic program $P = \{p \leftarrow a\,; p \leftarrow b\,; a \leftarrow \neg b\,; b \leftarrow \neg a\}$. P has three strict stable classes. Figure 1 shows these stable classes pictorially: we draw an arc from interpretation I to interpretation J if $F_P(I) = J$. The strict stable classes of P are: $S_1 = \{\{p, a\}\}$, $S_2 = \{\{p, b\}\}$, $S_3 = \{\emptyset, \{p, a, b\}\}$. The strict stable classes correspond to the cycles in Figure 1.

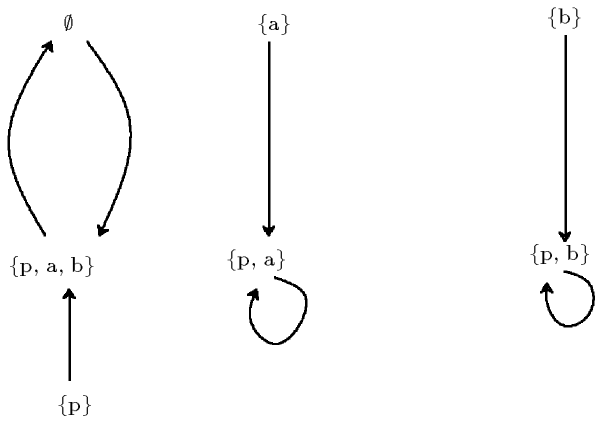

Figure 1: Graphical Representation: Program P

Theorem 1 *(Baral and Subrahmanian)* Every logic program P has at least one non-empty stable class. $\Box$

The following theorem proved in [BS90] will prove useful.

Theorem 2 [BS90] Suppose S_1, S_2 are strict stable classes of program P. Then $S_1 \cap S_2 \neq \emptyset$ iff $S_1 = S_2$. $\Box$

Theorem 2 basically tells us that if two strict stable classes share a common element, then they must be identical. Note that this doesn't necessarily hold for non-strict stable classes. However, in this paper, we are primarily concerned with strict stable classes only.

2.2 Well-founded vs Stable Class

Definition 2.4 Suppose I is an Herbrand interpretation and P is a logic program. Construct a program P' as follows: If

$$A \leftarrow B_1 \,\&\ldots\&\, B_n \,\&\, \neg D_1 \,\&\ldots\&\, \neg D_m$$

is in P, then

$$A \leftarrow B_1 \,\&\ldots\&\, B_n \,\&\, \tilde{D_1} \,\&\ldots\&\, \tilde{D_m}$$

is in P'. Here $\tilde{D_i} = \tilde{p}(\vec{t})$ iff $D_i = p(\vec{t})$. In addition, if $A \notin I$, then the unit clause $\tilde{A} \leftarrow$ is added to P'.

Now define $S_P(\overline{I}) = B_P \cap T_{P'} \uparrow \omega$.

Note that $T_{P'} \uparrow \omega$ contains new atoms of the form $\tilde{p}(\vec{t})$ and that these atoms may be used in deriving atoms in $S_P(\overline{I})$, but atoms of the form $\tilde{p}(\vec{t})$ are themselves not present in $S_P(\overline{I})$.

Lemma 1 $S_P(\overline{I}) = F_P(I)$

Proof

By definition, $S_P(\overline{I}) = B_P \cap T_{P'} \uparrow \omega$ and $F_P(I) = lfp(T_{P^I})$. To prove that $S_P(\overline{I}) = F_P(I)$, it suffices to prove by induction that, for all n, $B_P \cap T_{P'} \uparrow n = T_{P^I} \uparrow n$.

Base case : $B_P \cap T_{P'} \uparrow 0 = T_{P^I} \uparrow 0 = \emptyset$.

induction hypothesis : For all $m < n$, $B_P \cap T_{P'} \uparrow m = T_{P^I} \uparrow m$

induction step : To prove that $B_P \cap T_{P'} \uparrow n = T_{P^I} \uparrow n$.

$x \in B_P \cap T_{P'} \uparrow n$

$\Leftrightarrow x \in B_P$ and $x \in T_{P'} \uparrow n$

$\Leftrightarrow x \in B_P$ and There exists a rule $r : x \leftarrow a_1, \cdots, a_l, \tilde{b_1}, \cdots, \tilde{b_m}$ in P' such that $\{a_1, \cdots, a_l\} \subseteq T_{P'} \uparrow (n-1)$ and $\{\tilde{b_1}, \cdots, \tilde{b_m}\} \subseteq T_{P'} \uparrow (n-1)$

$\Leftrightarrow x \in B_P$ and There exists a rule $r : x \leftarrow a_1, \cdots, a_l, \neg b_1, \cdots, \neg b_m$ in P such that $\{a_1, \cdots, a_l\} \subseteq T_{P^I} \uparrow (n-1)$, and $\{b_1, \cdots, b_m\} \subseteq \overline{I}$.

$\Leftrightarrow x \in B_P$ and There exists a rule $r : x \leftarrow a_1, \cdots, a_l$ in P^I and $\{a_1, \cdots, a_l\} \subseteq T_{P^I} \uparrow (n-1)$.

$\Leftrightarrow x \in T_{P^I} \uparrow n$. □

Lemma 2 (Baral and Subrahmanian) Let P be any logic program. Then $F_P(lfp(F_P^2)) = gfp(F_P^2)$ and $F_P(gfp(F_P^2)) = lfp(F_P^2)$. i.e. $\{lfp(F_P^2), gfp(F_P^2)\}$ form a stable class of P. □

Definition 2.5 Given a logic program P, we associate an operator A_P with P as follows:
$A_P(I) = \overline{S_P(\overline{S_P(I)})}$.
$A^* = lfp(A_P)$.
$A^+ = S_P(A^*)$.

Well-founded semantics was initially defined by Van Gelder, Ross, and Schlipf [VRS88]. Later Van Gelder [Gel89] gave a different characterization of the well-founded semantics based on an alternating fixpoint approach. The following theorem formalizes Van Gelder's alternating fixpoint characterization.

Theorem 3 [Gel89] A^+ is the set of atoms true in the well-founded model of P. Similarly A^* is the set of atoms false in the well-founded model of P.

We now use Van Gelder's alternating fixpoint characterization to show that well-founded semantics is equivalent to a particular stable class.

Lemma 3 $lfp(F_P^2) = A^+$
$gfp(F_P^2) = \overline{A^*}$

Proof
$A_P(\overline{I}) = \overline{S_P(\overline{S_P(\overline{I})})}$.
$= \overline{S_P(\overline{F_P(I)})}$ (by Lemma 1)
$= \overline{F_P(F_P(I)))}$ (by Lemma 1)
$= HB - F_P(F_P(I))$
$= HB - F_P^2(I)$
It follows from the above that I is a fixpoint of A_P iff $\overline{I}$ is a fixpoint of F_P^2. It now follows immediately that $lfp(F_P^2) = \overline{gfp(A_P)}$ and $gfp(F_P^2) = \overline{lfp(F_P^2)}$. The second equality of the lemma has been established.

To establish the first equality, observe that

A^+
$= S_P(lfp(A_P))$
$= F_P(\overline{lfp(A_P)})$
$= F_P(gfp(F_P^2))$
$= lfp(F_P^2)$ (by Lemma 2). □

Lemmas 1, 2, 3 and Theorem 3 are required to establish the following theorem. (This result was established during discussions with David Scott Warren).

Theorem 4 (Well-Founded Semantics is Captured by a Stable Class)[1] Let P be any normal logic program. The well-Founded semantics of P is characterized by a particular stable class C of P, i.e. a ground atom is true in the well-founded semantics of P iff A is true in all interpretations in C, and A is false according to the well founded semantics of P iff A is false in all interpretations in C. Moreover, $C = \{lfp(F_P^2)\ ,\ gfp(F_P^2)\}$.

Proof

Consider $lfp(F_P^2)$ and $gfp(F_P^2)$. The set $\{lfp(F_P^2), gfp(F_P^2)\}$ is a stable class. $lfp(F_P^2) = A^+$ = the set of atoms true in the well-founded model of P. $\overline{gfp(F_P^2)} = A^*$ = the set of atoms false in the well-founded model of P. $\square$

Note that S_P, A_P, A^+, and A^* are used only as technical devices in proving Theorem 3. They are not needed in the statement of Theorem 3, which is expressed solely in terms of F_P, the Gelfond-Lifschitz operator.

2.3 Duality

Definition 2.6 Smyth Ordering

Suppose $(X, \leq)$ is a partially ordered set and S_1, S_2 are subsets of X. We say $S_1 \leq_s S_2$ iff $(\forall c \in S_1)\ (\exists d \in S_2)$: $c \leq d$

Definition 2.7 Hoare Ordering

Suppose $(X, \leq)$ is a partially ordered set and S_1, S_2 are subsets of X. We say $S_1 \leq_h S_2$ iff $(\forall d \in S_2)\ (\exists c \in S_1)$: $c \leq d$

In general $\leq_s$ and $\leq_h$ are not partial orders. But when the ordering is defined with respect to strict stable classes, then these orderings turn out to be partial orderings. Note that the set of Herbrand interpretations is a complete lattice under inclusion. Thus, we can define $\subseteq_h$ and $\subseteq_s$ on *sets* S_1, S_2 of interpretations as follows: $S_1 \subseteq_s S_2$ iff for every interpretation $I_1 \in S_1$, there is an interpretation $I_2 \in S_2$ such that $I_1 \subseteq I_2$. Likewise, $S_1 \subseteq_h S_2$ iff for every interpretation $I_2 \in S_2$, there is an interpretation $I_1 \in S_1$ such that $I_1 \subseteq I_2$. It turns out that when we restrict S_1, S_2 to be strict stable classes, then $\subseteq_h$ and $\subseteq_s$ are partial orderings.

[1] Credit for this theorem should really go to Van Gelder [Gel89] who proved Theorem 3. This theorem is essentially a re-statement of Theorem 3 which uses only one operator, viz. the familiar Gelfond-Lifschitz operator F_P, instead of two operators that van Gelder used.

Lemma 4 Suppose P is a logic program. Suppose S_1, S_2 are strict stable classes of P.

1. If $S_1 \subseteq_s S_2$ and $S_2 \subseteq_s S_1$, then $S_1 = S_2$.
2. If $S_1 \subseteq_h S_2$ and $S_2 \subseteq_h S_1$, then $S_1 = S_2$.

Proof:

(1) Suppose $S_1 \subseteq_s S_2$ and $S_2 \subseteq_s S_1$. Consider S_1. We can write S_1 as a disjoint union $Z_1 \cup \cdots \cup Z_k$ for some integer $k \geq 1$ as follows:

- $Z_1 = \{I \in S_1 \mid$ there is no $J \in S_1$ such that $J \subset I\}$ and
- $Z_{i+1} = \{I \in S_1 \mid$ there is no $J \in (S_1 - \bigcup_{j=1}^{i} Z_j)$ such that $J \subset I\}$.
- $Z_k \neq \emptyset$.

Thus, Z_1 contains the $\subseteq$-minimal interpretations in S_1, Z_2 contains the $\subseteq$-minimal interpretations in $(S_1 - Z_1)$, and Z_k contains the $\subseteq$-maximal elements of S_1. By a suitable choice of k, it is easy to see that we can arrange for Z_k to be non-empty.

Consider an $I \in Z_k$. Then, as $S_1 \subseteq_s S_2$, there exists a $J \in S_2$ such that $I \subseteq J$. Hence, as $S_2 \subseteq_s S_1$, there is an $H \in S_1$ such that $J \subseteq H$. But then

$$I \subseteq J \subseteq H.$$

But I is a $\subseteq$-maximal element of S_1. Thus, $I \subseteq H$ is possible iff $I = H$.

Hence, as $I = H$, and as $I \subseteq J \subseteq H$, $J = H = I$. Therefore, $I \in S_2$, and hence, $I \in S_1 \cap S_2$. It now follows by Theorem 2 that $S_1 = S_2$.

(2) The proof is analogous. The only difference this time is that we look at Z_1 and $\subseteq$-minimal elements, instead of looking at Z_k and consider $\subseteq$-maximal elements. □

Lemma 5 Suppose P is a logic program. Let $SSC(P)$ be the set of all strict stable classes of P. Then $(SSC(P), \subseteq_h)$ and $(SSC(P), \subseteq_s)$ are partially ordered sets.

Proof:

That $\subseteq_h$ and $\subseteq_s$ are reflexive and transitive is immediate. Anti-symmetry follows from Lemma 4. □

Definition 2.8 Cardinality Ordering
Suppose $(X, \leq)$ is a partially ordered set and S_1, S_2 are subsets of X. We say $S_1 \leq_c S_2$ iff the cardinality of S_1 is less than or equal to the cardinality of S_2.

In [BS90] we define the stable class semantics of program P to be the class of $\leq_s$-minimal strict stable classes of the program P. Theorem 5 below shows the duality between the stable class semantics and the well-founded semantics by showing that well-founded semantics of a program P is equivalent to the $\leq_h$ minimal strict stable class of P. Before proving this theorem, we need to define two properties about ordinals. Note that any ordinal γ may be written as the sum of a limit ordinal α, and a non-negative *integer* i. Given an ordinal α, we use $up(\alpha)$ to denote the smallest *limit* ordinal greater than or equal to α. Likewise, $down(\alpha)$ is the largest *limit* ordinal less than or equal to α. Thus, an ordinal α is a limit ordinal iff $up(\alpha) = \alpha = down(\alpha)$. As an example, $up(\omega + 5) = \omega 2$, while $down(\omega + 5) = \omega$. $up(0) = down(0) = 0$, $up(\omega) = down(\omega) = \omega$.

Upward iteration of monotone operators is defined in the usual way. For any function f mapping Herbrand interpretations to Herbrand interpretations, we set:

$f \uparrow 0 = \emptyset$
$f \uparrow (\alpha + 1) = f(f \uparrow \alpha)$ where $(\alpha + 1)$ is a successor ordinal
$f \uparrow \beta = \bigcup_{\alpha < \beta} f \uparrow \alpha$ for limit ordinals β

Theorem 5 Consider a logic program P. For any stable class S of P, $\{lfp(F_P^2), gfp(F_P^2)\} \leq_h S$, i.e. P has $\{lfp(F_P^2), gfp(F_P^2)\}$ as its $\leq_h$-minimal stable class.

Proof
Let S be a strict stable class of P. Hence, there exists a least integer $n > 0$ such that for all $x \in S$, x is a fixpoint of F_P^n.

Case 1. [Let n be even, i.e. $n = 2k$ for $k \geq 1$.] Clearly, any fixpoint of F_P^2 is also a fixpoint of F_P^{2k}, and hence $lfp(F_P^{2k}) \subseteq lfp(F_P^2)$.

We now need to show that $lfp(F_P^2) \subseteq lfp(F_P^{2k})$.

Claim 1: We first show that for all ordinals γ, $F_P^2 \uparrow \gamma \subseteq F_P^{2k} \uparrow up(\gamma)$.

Proof of Claim 1: First, observe that F_P^2 and F_P^{2k} are both monotone. We proceed by transfinite induction on γ.

Base Case. ($\gamma = 0$) In this case, $F_P^2 \uparrow \gamma = \emptyset = F_P^{2k} \uparrow \gamma$, and we are done.

Inductive case. ($\gamma > 0$) If γ is a limit ordinal, then $X \in F_P^2 \uparrow \gamma$ implies that $X \in F_P \uparrow \beta$ for some $\beta < \gamma$. But then, by the induction hypothesis, $X \in F_P^{2k} \uparrow up(\beta)$. As $\beta < \gamma$, and as γ is a limit ordinal, $up(\beta) \leq \gamma$; hence, by monotonicity of F_P^{2k}, it follows that $X \in F_P^{2k} \uparrow \gamma$.

Suppose on the other hand that γ is a successor ordinal, i.e. $\gamma = down(\gamma) + \ell$ where ℓ is an integer greater than zero. (If ℓ was zero, then γ would be a limit ordinal). As F_P^2 is a monotonic operator,

$$F_P^2 \uparrow (down(\gamma) + \ell) \subseteq F_P^2 \uparrow (down(\gamma) + k \times \ell).$$

By the induction hypothesis,

$$F_P^2 \uparrow down(\gamma) \subseteq F_P^{2k} \uparrow down(\gamma).$$

Thus,

$$\begin{aligned}
& F_P^2 \uparrow (down(\gamma) + k \times \ell) \\
& = \underbrace{F_P^2 \cdots F_P^2}_{k \times \ell \text{ times}}(F_P^2 \uparrow down(\gamma)) \\
& = \underbrace{F_P^{2k} \cdots F_P^{2k}}_{\ell \text{ times}}(F_P^2 \uparrow down(\gamma)) \\
& \subseteq \underbrace{F_P^{2k} \cdots F_P^{2k}}_{\ell \text{ times}}(F_P^{2k} \uparrow down(\gamma) \\
& \subseteq F_P^{2k} \uparrow up(\gamma).
\end{aligned}$$

The second last inclusion follows by monotonicity of F_P^{2k} and as

$$F_P^2 \uparrow down(\gamma) \subseteq F_P^{2k} \uparrow down(\gamma).$$

This completes the proof of Claim 1.

At this point, we have shown that when S is a strict stable class of even cardinality $n = 2k$, it is the case that whenever $X \in F_P^2 \uparrow \gamma$, it must be the case that $X \in F_P^{2k} \uparrow up(\gamma)$. As F_P^2 and F_P^{2k} are both monotone, there are ordinals α and β such that $lfp(F_P^2) = F_P^2 \uparrow \alpha$ and $lfp(F_P^{2k}) = F_P^{2k} \uparrow \beta$. Let $\gamma = up(max\{\alpha, \beta\})$, i.e. γ is the smallest limit ordinal greater than both α, β. Clearly, $lfp(F_P^2) = F_P^2 \uparrow \gamma$ and $lfp(F_P^{2k}) = F_P^{2k} \uparrow \gamma$. But, by Claim 1, $F_P^2 \uparrow \gamma \subseteq F_P^{2k} \uparrow \gamma$ as γ is a limit ordinal. Hence, $lfp(F_P^2) \subseteq lfp(F_P^{2k})$.

This completes the proof that $lfp(F_P^2) \subseteq lfp(F_P^n)$ for $n = 2k$. As we already know that $lfp(F_P^n) \subseteq lfp(F_P^2)$, we have now finished showing that $lfp(F_P^2) = lfp(F_P^n)$ for $n = 2k$.

As every strict stable class S of even cardinality must have $k \geq 1$, it follows that for each $x \in S$, $x \subseteq lfp(F_P^2)$ (as x is a fixpoint of F_P^n). Hence, $\{lfp(F_P^2), gfp(F_P^2)\} \leq_h S$ where S is a strict stable class of even cardinality. This completes case 1.

Case 2. [Let n be odd]. It is obvious that any fixpoint of F_P^n is also a fixpoint of F_P^{2n}. Since, $2n$ is even, for all elements x in S, $lfp(F_P^2) \subseteq x$.

Since, a stable class is a union of strict stable classes, for any stable class S of P, and any element x in S, $lfp(F_P^2) \subseteq x$. Hence, $\{lfp(F_P^2), gfp(F_P^2)\} \leq_h S$. $\square$

Definition 2.9 Given a logic program P we denote the stable class $\{lfp(F_P^2), gfp(F_P^2)\}$ by $CAN^{SC}(P)$.

Corollary 2.2 Suppose P is a logic program. $CAN^{SC}(P)$ is equivalent to the well-founded semantics of P in the sense that:

1. a ground atom A is true in the well-founded semantics of P iff $A \in lfp(F_P^2)$ and
2. a ground atom A is false in the well-founded semantics of P iff $A \notin gfp(F_P^2)$

Corollary 2.3 The well-founded semantics of P is two-valued iff $lfp(F_P^2) = gfp(F_P^2)$; i.e. F_P^2 has a unique fixpoint. $\square$

Corollary 2.4 Suppose P is a normal logic program with a unique stable model I. Then the well-founded semantics for P is not two-valued iff:

1. $lfp(F_P^2) \neq gfp(F_P^2)$ and
2. $I \neq lfp(F_P^2)$. $\square$

It has been known for some time that there exist programs P that have a unique stable model, but for which the well-founded semantics is not two valued. The above corollary explains precisely under what circumstances this phenomenon occurs. For a concrete example of such programs, consider the program $P = \{q \leftarrow \neg r\ ,\ r \leftarrow \neg q\ ;\ p \leftarrow \neg p\ ;\ p \leftarrow \neg r\ \}$ [VRS88]. P has a unique stable model, viz. $\{p, q\}$. Figure 2 shows that P has three strict stable classes C_1, C_2 *and* C_3 where $C_1 = \{\{q, p\}\}$, $C_2 = \{\emptyset, \{p, q, r\}\}$ and $C_3 = \{\{r\}, \{r, p\}\}$. Of these, the class C_2 corresponds to the well-founded semantics which says that p, q, r are all unknown.

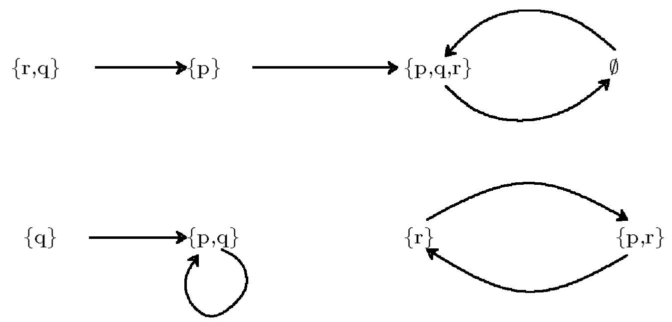

Figure 2: Graphical Representation of Stable Classes

We showed that the well-founded semantics is based on the Hoare ordering and the stable class semantics is based on the Smyth ordering. We observe that the stable model semantics is based on the Cardinality ordering.

If a program has stable models then the semantics based on the cardinality ordering is equivalent to the stable model semantics. If a program does not have a stable model then the $\subseteq_c$-minimal classes based on the cardinality ordering includes the class corresponding to the well-founded semantics of the program. Note, however, that there may be other strict stable classes of cardinality 2.

3 Default theories

Default logic introduced by Reiter [Rei80] forms one of the most important formalisms for non-monotonic reasoning. Recently, the relationship between default logics and other forms of non-monotonic reasoning have been studied intensely (in particular the work of Konolige [Kon88] and the work of Marek and Truszczynski [MT89a, Mar89] show that default and auto-epistemic logics are not that different after all). More recently, the relationship between logic program semantics and non-monotonic logics has been studied by Marek and

Truszczynski [MT89b] and by Marek and Subrahmanian [MS88]. In all these cases, correspondences are established between certain structures called extensions in default logic, and certain classes of models of logic programs. However, default theories may not always have extensions. In such cases, these correspondences are not very meaningful.

In [BS90] we introduced the concept of *extension class* for default theories. All default theories possess non-empty extension classes. In [BS90] we also showed various correspondences between extension classes and stable classes of logic programs. We recapitulate the main results from [BS90] in Section 3.1.

We assume that we have a fixed language L, whose ground atoms are denoted by AT. A (propositional) default is a triple $d = (p(d), j(d), c(d))$, where $p(d)$ and $c(d)$ are formulas of $L = L(AT)^2$, and $j(d)$ is a finite subset of L. $p(d)$ is called the *prerequisite* of d, $j(d)$ the *justification* of d, and $c(d)$ the *consequent*, or *conclusion* of d. Default d is usually denoted by $\frac{p(d):j(d)}{c(d)}$. A default theory is a pair (D, W), where $W \subseteq L$ and D is a collection of defaults.

Given a default theory (D, W) and a set E of formulas (called the *context*), we can define an operator $R^{E,D}$ which maps sets of formulas to sets of formulas in the following way:

$$R^{E,D}(S) = Cn(S \cup \{c(d) : d \in D, p(d) \in S, \neg j(d) \cap E = \emptyset\}).$$

Here, Cn denotes the familiar Tarskian consequence operator and $\neg j(d)$ denotes the set $\{\neg\beta \mid \beta \in j(d)\}$. However, we abuse notation in the following way: if I is an interpretation, we also use $Cn(I)$ to denote the set of all formulas satisfied by the interpretation I. The operator $R^{E,D}(S)$ is related to the notion of strong default proof discussed in [MT89a]. Starting with $Cn(W)$, the least fixpoint of this operator above $Cn(W)$ is the set of formulas that have a strong proof [MT89a] from W using D and with respect to the context E. More formally,

Definition 3.1 Suppose (D, W) is a default theory, and E is a set of formulas, called the *context*. Then:

$R_0^{E,D}(W) = Cn(W)$.
$R_{n+1}^{E,D}(W) = R^{E,D}(R_n^{E,D}(W))$
$R_\infty^{E,D}(W) = \bigcup_{n=0}^{\infty} R_n^{E,D}(W)$ □

Note (cf. [MT89b]) that if $\Delta = (D, W)$ is a default theory, $R_\infty^{E,D}(W)$ is identical to $\Gamma_\Delta(E)$

[2] $L(AT)$ is the propositional language generated by considering the ground atoms in AT to be propositional symbols.

where Γ_Δ is the operator associated with default theories originally defined by Reiter in [Rei80].

Example 3.1 Suppose (D,W) is the default theory

$$W = \{p \rightarrow q\} \ , \ D = \{\frac{q : \neg w}{p} \ , \ \frac{: \neg s}{q}\}$$

and E is the context $E = Cn(\{r\})$. Then:

$R_0^{E,D}(W) = Cn(W)$
$R_1^{E,D}(W) = Cn(W \cup \{q\})$
$R_2^{E,D}(W) = Cn(W \cup \{q, p\}) = R_\infty^{E,D}(W)$ □

The most important structures of default logic are extensions, which are formally defined as follows.

Definition 3.2 E is an extension of a default theory (D,W) iff $E = R_\infty^{E,D}(W)$. □

If a particular default theory (D,W) has a unique extension then the meaning of that theory is considered to be the set of formulas present in the extension. Note that extensions are always closed under propositional consequence.

However, many default theories may have multiple extensions or no extensions at all. Our goal is to explore the meaning of the default theory (D,W) in such a case.

Example 3.2 Consider the default theory (D,W), where $W = \{p\}$ and $D = \{\frac{:\neg q}{q}\}$. The two possible extensions are:
$E_1 = Cn(\{p\})$ and $E_2 = Cn(\{p,q\})$.
But $R_\infty^{E_1,D} = Cn(\{p,q\}) \neq E_1$ and also $R_\infty^{E_2,D} = Cn(\{p\}) \neq E_2$. Hence, neither E_1 nor E_2 are extensions of this default theory and hence this default theory has no extensions. It seems clear that p should be true in this default theory (and of course, $\neg p$ should not be true in this theory). Hence, we feel that extensions are not strong enough to capture the meaning of a default theory. □

Extension Classes, defined below, are able to handle the semantics of default logics when they have no extensions.

Definition 3.3 A *finite* family, $E = (E_i)_{i \in \mathcal{A}}$ of sets of formulas is an *extension class* of (D,W) iff
$E = \{R_\infty^{E_i,D}(W) \mid E_i \in E.\}$

As defined above an extension class is a set of *sets of formulas*. A formula F is assigned *true* (resp. *false*) by an extension class $E = (E_i)_{i \in \mathcal{A}}$ of a default theory (D, W) iff F is true (resp. false) in each E_i, $i \in \mathcal{A}$.

Example 3.3 We again consider the default theory (D, W), where $W = \{p\}$ and $D = \{\frac{:\neg q}{q}\}$. This default theory has an extension class

$$\{Cn(\{p\}), Cn(\{p, q\})\}.$$

Since, p is true in all sets of formulas of this extension class we can assume p to be true in the extension class semantics of this default theory. □

Theorem 6 *(Baral and Subrahmanian)* Suppose AT is a finite set of ground atoms. Every propositional default theory has a non-empty extension class.

3.1 Relationship between Stable Classes and Extension Classes

Marek and Truszczynski [MT89b] present three different translation of logic programs to default logic and show the relationship between the stable models of the original program and the extensions of the default theories generated by the different translations of the original program. Keeping with their terminology two of the transformations are :

tr_1 : Let $C = a \leftarrow b_1 \,\&\, \cdots b_k \,\&\, \neg c_1 \,\&\, \cdots \,\&\, \neg c_r$.
$tr_1(C) = \frac{b_1 \wedge \cdots \wedge b_k : \neg c_1, \cdots, \neg c_r}{a}$.
If P is a program $tr_1(P) = (D_1(P), W_1(P))$ is a default theory corresponding to P, where $W_1(P) = \emptyset$ and $D_1(P) = \{tr_1(C) : C \in P\}$.[3]

tr_2 : Let $C = a \leftarrow b_1 \cdots b_k, \neg c_1 \cdots \neg c_r$.
$tr_2(C) = \frac{:\neg c_1 \cdots \neg c_r}{b_1 \wedge \cdots \wedge b_k \Rightarrow a}$.
If P is a program $tr_2(P) = (D_2(P), W_2(P))$ is a default theory corresponding to P, where $W_2(P) = \emptyset$ and $D_2(P) = \{tr_2(C) : C \in P\}$.

In [BS90] we show the relationship between stable classes of a program, P and the extension classes of the translation of P. The following two theorems formalize these relationships.

Theorem 7 [BS90] M is a stable class of a program P iff $E = \{Cn(M_i) \mid M_i \in M\}$ is an extension class of $tr_1(P)$. □

[3] tr_1 was first introduced in [BF88].

Theorem 8 [BS90] Suppose M is a collection of interpretations. M is a stable class of program P iff there exists an extension class E of $tr_2(P)$, such that $M = \{E_i \cap AT \mid E_i \in E\}$ is an extension class of $tr_2(P)$. $\Box$

3.2 Well-founded semantics for default theories

We now show how the result of [BS90] allow us to obtain a well-founded semantics for default theories. Consider a default theory $T = (D, W)$. To make the definition uniform, let $R_{\infty}^{E,D}(W)$ be denoted as $\mathcal{F}_T(E)$. It can easily be proved that $\mathcal{F}_T$ is anti-monotonic and hence $\mathcal{F}_T^2$, the function that applies $\mathcal{F}_T$ twice is monotonic. We now define the well-founded semantics of a default theory based on the greatest and least fixpoint of $\mathcal{F}_T^2$.

Definition 3.4 (Well-founded semantics for default theories) A formula $\mathcal{F}$ is *true* in a default theory T with respect to the well-founded semantics if $\mathcal{F} \in lfp(\mathcal{F}_T^2)$. $\mathcal{F}$ is *false* in T with respect to the well-founded semantics if $\mathcal{F} \notin gfp(\mathcal{F}_T^2)$. Otherwise $\mathcal{F}$ is said to be *unknown* w.r.t. T.

Since, $\mathcal{F}_T^2$ is monotonic the least fixpoint of $\mathcal{F}_T^2$ can be computed by iterating $\mathcal{F}_T^2$ upwards, starting from the empty set until it reaches a fixpoint. Similarly, the greatest fixpoint of $\mathcal{F}_T^2$ can be computed by iterating $\mathcal{F}_T^2$ downwards, starting from the set of all formulas, until it reaches a fixpoint. Such an iteration may well be transfinite.

Theorem 9 (Relation between well-founded semantics of logic programs and default theories) Let P be a normal logic program and $tr_1(P)$ and $tr_2(P)$ be its translations to a default theory.
1. A ground atom X is *true* (*false*) in the well-founded semantics of P iff X is *true* (*false*) in the well-founded semantics of $tr_1(P)$.
2. An atom X is *true* (*false*) in the well-founded semantics of P iff X is *true* (*false*) in the well-founded semantics of $tr_2(P)$.

Proof:
Directly from Theorems 7 and 8. $\Box$

We now discuss the role of the various orderings on the strict extension classes. In [BS90] we define the extension class semantics to be the union of $\leq_s$ minimal strict extension classes. We now show the duality between the extension class semantics and the well-founded semantics by showing that well-founded semantics is equivalent to the unique $\leq_h$ minimal strict extension class.

Theorem 10 Consider a default theory T. For any extension class E of T, $\{lfp(\mathcal{F}_T^2), gfp(\mathcal{F}_P^2)\} \leq_h E$, i.e. T has $\{lfp(\mathcal{F}_T^2), gfp(\mathcal{F}_T^2)\}$ as its $\leq_h$-minimal extension class. □

For coherent default theories [4] the set of minimal extension classes based on the cardinality ordering correspond to the set of extensions of the default theory. For incoherent default theories the set of minimal extension classes based on the cardinality ordering includes the extension class corresponding to the well-founded semantics of the default theory.

3.3 Weak Extension Classes and Stable Classes

Weak extensions of default theories were introduced by Marek and Truszczynski [MT89b] and were proven to be essentially equivalent to expansions in auto-epistemic logic. In [BS90], we defined the concept of weak extension classes and study its relationship with stable classes.

Given a set D of defaults, we write $c(D)$ to denote the set $\{c(d) \mid d \in D\}$, i.e. $c(D)$ is the set of all conclusions of defaults occurring in D.

Definition 3.5 *[MT89b]* E is said to be a *weak extension* of a default theory (D, W) iff $E = Cn(W \cup \{c(d) \mid \frac{p(d)\,:\,j(d)}{c(d)}$ is a default in D and $p(d) \in E$ and $\neg j(d) \cap E = \emptyset\})$.

Suppose $T = (D, W)$ is a default theory and E is a set of wffs. We associate an operator G_T with T that maps sets of wffs to sets of wffs. G_T is defined as follows: $G_T(E) = Cn(W \cup \{c(d) \mid \frac{p(d)\,:\,j(d)}{c(d)}$ is a default in D and $p(d) \in E$ and $\neg j(d) \cap E = \emptyset\})$.

Definition 3.6 A finite collection $(E_i)_{i \in \mathcal{A}}$ of sets of formulas is said to be a *weak extension class* of the default theory (D, W) iff $\{E_i \mid i \in \mathcal{A}\} = \{G_T(E_j) \mid j \in \mathcal{A}\}$.

The function G_T is not antimonotonic. The following example illustrates this.

Example 3.4 Let T be the default theory (D, W) where, $D = \{\frac{a:b}{c}, \frac{b:}{d}\}$ and $W = \emptyset$. Let $E_1 = \{a\}$ and $E_2 = \{a, b\}$. $G_T(E_1) = \{c\}$ and $G_T(E_2) = \{d\}$. Therefore, $G_T(E_2) \not\subseteq G_T(E_1)$. Hence, G_T is not antimonotonic.

The above example can easily be extended to show that G_T^2, the function obtained by applying G_T twice is not monotonic. Hence, the least and greatest fixpoint of G_T^2 can not

[4] i.e. Default theories that have at least one extension.

be obtained by an iterative approach. This greatly complicates the study of class theory for weak extensions, as well as for auto-epistemic logic. We show below how class theory for strong auto-epistemic logic, the logic of iterative expansions introduced by Marek and Truszczynski [MT89b], may be studied.

4 Autoepistemic theories

Autoepistemic logic was introduced by Moore [Moo85]. Its relationship with logic program semantics has been extensively studied in the literature [GL88, MS88, MT89b]. Its relationship with other non-monotonic logics like default logic [MS88] and circumscription [Prz89] has also been studied.

4.1 Expansions in auto-epistemic logic

Technically, autoepistemic logic is obtained by extending a first order language L, with a modal operator K. The new language, L_K, is obtained by requiring that if ϕ is a wff of L, then ϕ and $K\phi$ are wffs of L_K. The definition of L_K is then closed under application of connectives in the usual way. Intuitively, $K\phi$ is to be read as "ϕ is known to be true" or "ϕ appears on the list of sentences accepted by the reasoning agent as known". The meaning of theories in L_K is usually characterized by a construct called an *expansion*.

Definition 4.1 Let $A \subseteq L_K$ be an autoepistemic theory. E is an expansion of A iff
$E = Cn(A \cup \{K\phi : \phi \in E\} \cup \{\neg K\psi : \psi \notin E\})$

Note that here, we treat $K\phi$ as a new atom as do Marek and Truszczynski [MT89b] and Konolige [Kon88], and hence, the Cn consequence relation refers to classical logical consequence.

Given an auto-epistemic theory A, we associate an operator $\mathcal{I}_A$ that maps sets of wffs (of L_K) to sets of wffs of L_K as follows: $\mathcal{I}_A(E) = Cn(A \cup \{K\phi : \phi \in E\} \cup \{\neg K\psi : \psi \notin E\})$.

The function $\mathcal{I}_A$ is not antimonotonic. The following example illustrates this.

Example 4.1 Let A be the autoepistemic theory $\{a, b, Kc\}$. Let $E_1 = \{a\}$ and $E_2 = \{a, b\}$. Then, $\mathcal{I}_A(E_2)$ contains Kb while $\mathcal{I}_A(E_1)$ does not contain Kb.Therefore, $I_A(E_2) \not\subseteq I_A(E_1)$. Hence, $\mathcal{I}_A$ is not antimonotonic.

The above example can be easily extended to show that $\mathcal{I}_A^2$ is not monotonic. Hence, we cannot define well-founded semantics for autoepistemic theories based on least and greatest fixpoints of $\mathcal{I}_A^2$. This is the key distinction between class theory for default logic and stable semantics on the one side, and auto-epistemic logic on the other.

4.2 Weak Well-Founded Semantics for Autoepistemic theories

We now consider the function used by Marek and Truszczynski [MT89b] in defining iterative expansions of autoepistemic theories and define the concept of weak well-founded semantics for autoepistemic theories.

Definition 4.2 (Iterative Expansion) [MT89b] Let $B(S) = Cn(S \cup KS)$.
$B_0^E(A) = Cn(A \cup \neg K\overline{E})$
$B_{n+1}^E(A) = B(B_n^E(A))$
$B^E(A) = \bigcup_{0 \leq n < \omega} B_n^E(A)$
E is an iterative expansion of A iff $E = B^E(A)$.

Given an auto-epistemic theory, we may associate an operator G_A which maps sets of wffs of L_K to sets of wffs of L_K as follows: $G_A(E) = B^E(A)$.

Definition 4.3 A finite, non-empty set $\mathbf{E}$ of sets of wffs of L_K is said to be an *iterative expansion class* for an auto-epistemic theory A iff $\mathbf{E} = \{G_A(E_i) \mid E_i \in \mathbf{E}\}$.

Theorem 11 G_A is antimonotonic.

Proof :
If $E_1 \subseteq E_2$, $\neg K\overline{E_2} \subseteq \neg K\overline{E_1}$
Hence, $A \cup \neg K\overline{E_2} \subseteq A \cup \neg K\overline{E_1}$
Hence, as propositional consequence is monotonic, $Cn(A \cup \neg K\overline{E_2}) \subseteq Cn(A \cup \neg K\overline{E_1})$
Hence $B_0^{E_2}(A) \subseteq B_0^{E_1}(A)$. ————(1)

Consider $S_1 \subseteq S_2$, this implies $KS_1 \subseteq KS_2$
Hence, $S \cup KS_1 \subseteq S \cup KS_2$
Hence, $Cn(S \cup KS_1) \subseteq Cn(S \cup KS_2)$
Hence, $B(S_1) \subseteq B(S_2)$. ————(2)

Consider S_1 to be $B_0^{E_2}(A)$ and S_2 to be $B_0^{E_1}(A)$.
From (1) $S_1 \subseteq S_2$ is true. Hence, using (2) $B(B_0^{E_2}(A)) \subseteq B(B_0^{E_1}(A))$
By monotonicity of the operator B we have, $\bigcup_{0 \leq n < \omega} B_n^{E_2}(A) \subseteq \bigcup_{0 \leq n < \omega} B_n^{E_1}(A)$
Hence $B^{E_2}(A) \subseteq B^{E_1}(A)$
$G_A(E_2) \subseteq G_A(E_1)$, i.e. G_A is antimonotonic. □

Hence, G_A^2 is always monotonic, and we may define the weak well-founded semantics for auto-epistemic logic as follows:

Definition 4.4 (Weak Well-founded semantics for Autoepistemic theories) A formula $F \in L_K$ is *true* in an autoepistemic theory A with respect to the weak well-founded semantics if $F \in lfp(G_A^2)$. A formula $F \in L_K$ is *false* in an autoepistemic theory A with respect to the weak well-founded semantics if $F \notin gfp(G_A^2)$. Otherwise F is said to be *unknown*.

We now discuss the role of the various orderings on the strict iterative expansion classes. The iterative expansion class semantics is defined to be the class of $\leq_s$ minimal strict iterative expansion classes. We now show the duality between the iterative expansion class semantics and the weak well-founded semantics by showing that weak well-founded semantics is equivalent to the unique $\leq_h$ minimal strict iterative expansion class.

Theorem 12 Consider an autoepistemic theory A. For any iterative expansion class E of A,

$$\{lfp(G_A^2), gfp(G_A^2)\} \leq_h E$$

i.e. A has $\{lfp(G_A^2), gfp(G_A^2)\}$ as its $\leq_h$-minimal iterative expansion class. □

For autoepistemic theories that have at least one iterative expansion, the set of minimal iterative expansion classes based on the cardinality ordering correspond to the set of iterative expansion of the autoepistemic theory. For autoepistemic theories that do not have iterative expansions, the set of minimal iterative expansion classes based on the cardinality ordering includes the iterative expansion class corresponding to the weak well-founded semantics of the autoepistemic theory.

4.3 Tying together Classes in Auto-Epistemic Logic and Stable Classes

Marek and Truszczynski [MT89b] present a translation of logic programs to autoepistemic theories, which we call *epi*, and show the relationship between the stable models of the original program and the expansions and iterative expansions of the auto-epistemic theories generated by the different translations of the original program.

epi: Let $C = A \leftarrow B_1 \,\&\, \cdots B_k \,\&\, \neg C_1 \,\&\, \cdots \,\&\, \neg C_r$.
$epi(C) = \neg K C_1 \wedge \cdots \wedge \neg K C_r \Rightarrow (B_1 \wedge \cdots \wedge B_k \Rightarrow A)$.
For the program P, $epi(P) = \{epi(C) \mid C \in P\}$.

We know from [MT89b] that expansions and iterative expansions coincide for $epi(P)$.

Theorem 13 [MT89b] Let M be a collection of atoms. Then M is a stable model for a program C iff there exists an expansion A of $epi(P)$ such that $M = A \cap At$. $\Box$

Lemma 6 $F_P(I) = G_{epi(P)}(I) \cap At$

Proof:
Consider $A \leftarrow B_1, \cdots, B_k \in P^I$.
$\Leftrightarrow A \leftarrow B_1, \cdots, B_k, \neg C_1, \cdots, \neg C_r \in P$ and $\{C_1, \cdots, C_r\} \cap I = \emptyset$.
$\Leftrightarrow (A \Leftarrow B_1, \cdots, B_k) \Leftarrow \neg K C_1, \cdots, \neg K C_r \in epi(P)$ and $\{\neg K C_1, \cdots, \neg K C_r\} \subseteq \neg K \overline{I}$
$\Leftrightarrow (A \Leftarrow B_1, \cdots, B_k) \in B_0^I(epi(P))$.
Hence, $A \in F_P(I)$ iff $A \in B_0^I(epi(P))$ and A is an atom. By definition of $B^I(epi(P))$, and since $epi(P)$ does not have elements with K, $B_0^I(epi(P)) \cap At = B^I(epi(P)) \cap At$. Hence, $F_P(I) = G_{epi(P)}(I) \cap At$. $\Box$

Based on the above results, we are now in a position to state that:

Theorem 14 (Relationship with well-founded semantics of logic programs and weak well-founded semantics of auto-epistemic theories) Let P be a normal logic program and $epi(P)$ be its translations to a default theory.
An atom X is *true* (*false*) in the well-founded semantics of P iff X is *true* (*false*) in the weak well-founded semantics of $epi(P)$.

Proof:
Directly from Lemma 6. $\Box$

4.4 Kernel Expansions in Autoepistemic theories

It is known that each and every wff in auto-epistemic logic is logically equivalent to a wff of the "standard form":

$$K\phi_1 \wedge \cdots \wedge K\phi_m \wedge \neg K\psi_1 \wedge \cdots \wedge \neg K\psi_n \Rightarrow W$$

where $W \in L$ (and hence contains no occurrences of K) and each ϕ_i, $1 \leq i \leq m$ and ψ_j, $1 \leq j \leq n$ is a wff of L_K. Henceforth, we assume that all wff's are of the above standard form.

Consider level-one autoepistemic theories consisting of a set of clauses of the above standard form, with the added restriction that the ϕ_i's and ψ_j's are elements of the language L. For a level-one autoepistemic theory A, and $E \subseteq L$, we define a transformation $\mathcal{G}$ that takes A and E as input and produces a set of wffs of L (not L_K !) as output. $\mathcal{G}(A, E)$ is obtained from A as follows by considering each implication C in A of the form

$$K\phi_1 \wedge \cdots \wedge K\phi_m \wedge \neg K\psi_1 \wedge \cdots \wedge \neg K\psi_n \Rightarrow W.$$

If $\{\neg\phi_1, \ldots, \neg\phi_m\} \cap E = \emptyset$ and $\{\psi_1, \ldots, \psi_n\} \cap E = \emptyset$, then W is in $\mathcal{G}(A, E)$. Nothing else is in $\mathcal{G}(A, E)$.

Let $H_A(E)$ be defined as $Cn(\mathcal{G}(A, E))$. H_A is antimonotonic, and the function H_A^2 obtained by applying H_A twice is monotonic.

Definition 4.5 E is said to be a *kernel expansion* of a level-one autoepistemic theory A iff $E = H_A(E)$.

A collection $\mathbf{E}$ of *sets* of wffs of L_K is said to be a *kernel expansion class* of A iff $\mathbf{E} = \{H_A(E_i) \mid E_i \in \mathbf{E}\}$.

Definition 4.6 (Well-founded semantics for Level-one Autoepistemic theories) A formula $F \in L$ is *true* in a level-one autoepistemic theory A with respect to the well-founded semantics if $F \in lfp(H_A^2)$. A formula $F \in L$ is *false* in a level-one autoepistemic theory A with respect to the well-founded semantics if $F \notin gfp(H_A^2)$. Otherwise F is said to be *unknown*

Lemma 7 *Let P be a normal logic program, $epi(P)$ be its translation to an autoepistemic theory and I be an interpretation. $F_P(I) = H_{epi}(I) \cap At$*

Proof:
$A \leftarrow B_1, \cdots, B_k \in P^I$.
$\Leftrightarrow A \leftarrow B_1, \cdots, B_k, \neg C_1, \cdots, \neg C_r \in P$ and $\{C_1, \cdots, C_r\} \cap I = \emptyset$.
$\Leftrightarrow (A \Leftarrow B_1, \cdots, B_k) \Leftarrow \neg KC_1, \cdots, \neg KC_r \in epi(P)$ and $\{C_1, \cdots, C_r\} \cap I = \emptyset$.
$\Leftrightarrow (A \Leftarrow B_1, \cdots, B_k) \in \mathcal{G}(epi(P), I)$
Since, $H_{epi}(I) = Cn(\mathcal{G}(epi(P), I))$, $F_P(I) = H_{epi}(I) \cap At$. □

Theorem 15 (Relation between well-founded semantics of logic programs and autoepistemic theories) An atom X is *true* (*false*) in the well-founded semantics of P iff X is *true* (*false*) in the well-founded semantics of $epi(P)$.

Proof:
Directly from Lemma 7 $\square$

The various orderings (Smyth, Hoare and Cardinality) between strict expansion classes generate different semantics for level-one autoepistemic theories. The expansion class semantics is defined to be the union of $\leq_s$ minimal strict expansion classes.

Theorem 16 Consider an level-one autoepistemic theory A. For any expansion class E of A, $\{lfp(H_A^2), gfp(H_A^2)\} \leq_h E$, i.e. A has $\{lfp(H_A^2), gfp(H_A^2)\}$ as its $\leq_h$-minimal expansion class. $\square$

Proof:
Similar to proof of theorem 3. $\square$

For level-one autoepistemic theories that have at least one expansion, the set of minimal expansion classes based on the cardinality ordering correspond to the set of expansions of the level-one autoepistemic theory.

We now briefly discus nonmonotonic version of modal logics. Some of the nonmonotonic versions of modal logics have recently been proven to be equivalent to formulations of autoepistemic logic [SH90, MTS91]. All the modal logics we consider here have two inference rules: modus ponens and necessitation $\frac{\phi}{K\phi}$. Their axioms include all propositional tautologies in L_K and some axiom schemata that specify properties of the operator K. Among the most widely considered are the axiom schemas:

K $\quad K(\phi \Rightarrow \psi) \Rightarrow (K\phi \Rightarrow K\psi)$,

T $\quad K\phi \Rightarrow \phi$,

4 $\quad K\phi \Rightarrow KK\phi$,

5 $\quad \neg K \Rightarrow K\neg K\phi$.

The logic $\mathcal{N}$ is based on only the modus ponens and necessitation inference rules. By K we denote the modal logic based on the axiom **K**, T denotes the logic based on axioms **K** and **T**, S4 is the logic based on **K**, **T** and **4**, and S5 is based on **K**, **T**, **4** and **5**.

McDermott and Doyle [McD82] were the first ones to systematically study nonmonotonic reasoning within modal logic. Let $\mathcal{S}$ be a modal logic. By $Cn_{\mathcal{S}}$ we mean the consequence operator for modal logic $\mathcal{S}$, where $\mathcal{S}$ is any of the modal logics described above. McDermott and Doyle described a way to construct a nonmonotonic version of a given modal logic. They characterized the meaning of the nonmonotonic version of a given modal logic $\mathcal{S}$ using a construct called $\mathcal{S} - expansion$.

Definition 4.7 Let $A \subseteq L_K$ be a theory in modal logic $\mathcal{S}$. E is an $\mathcal{S} - expansion$ of A iff $E = Cn_{\mathcal{S}}(A \cup \{\neg K\psi : \psi \notin E\})$.

$\mathcal{S}$-expansion's suffer from the same problem as stable models and extensions. Hence, for a given theory A in a logic $\mathcal{S}$ we introduce the operator $\mathcal{F}_{\mathcal{S},A}$ which maps sets of wffs of L_K to sets of wffs of L_K as follows:
$\mathcal{F}_{\mathcal{S},A}(E) = Cn_{\mathcal{S}}(A \cup \{\neg K\psi : \psi \notin E\})$. Its easy to see that the fixpoints of $\mathcal{F}_{\mathcal{S},A}$ are $\mathcal{S}$-expansions of the theory A.

$\mathcal{S}$-expansion classes of A may be defined in the following way: A set $\mathbf{E}$ of sets of wffs is said to be a $\mathcal{S}$-expansion class of A iff $\mathbf{E} = \{\mathcal{F}_{\mathcal{S},A}(E) \mid E \in \mathbf{E}\}$.

Theorem 17 The function $\mathcal{F}_{\mathcal{S},A}(E)$ is antimonotonic, with respect to E, i.e. $E_1 \subseteq E_2 \Rightarrow \mathcal{F}_{\mathcal{S},A}(E_2) \subseteq \mathcal{F}_{\mathcal{S},A}(E_1)$. $\square$

Corollary 4.1 For all monotonic modal logics $\mathcal{S}$, $\mathcal{F}_{\mathcal{S},A}$ is antimonotonic with respect to E and hence the function $\mathcal{F}^2_{\mathcal{S},A}$ obtained by applying the function $\mathcal{F}_{\mathcal{S},A}$ twice is monotonic.

Definition 4.8 For any modal logic $\mathcal{S}$, where $\mathcal{F}^2_{\mathcal{S},A}$ is a monotonic function, the set $\{lfp(\mathcal{F}^2_{\mathcal{S},A}), gfp(\mathcal{F}^2_{\mathcal{S},A})\}$ exists and is called the $\mathcal{S}-$well-founded semantics.

The logic $\mathcal{N}$

The operator $\mathcal{F}_{\mathcal{N},A}$ is equivalent to the operator G_A defined in the last section [MTS91]. All other results related to iterative expansions, iterative expansion classes hold. We call the set $\{lfp(\mathcal{F}^2_{\mathcal{N},A}), gfp(\mathcal{F}^2_{\mathcal{N},A})\}$ as the $\mathcal{N}-$well-founded semantics.

The logic K45

Shvarts proves [SH90] that Moore's expansions coincide with K45-expansions. As noted in the last section the function defined by Moore to specify expansions is not antimonotonic and can not be used to define a well-founded semantics. But since $\mathcal{F}_{K45,A}$ is antimonotonic the the set $\{lfp(\mathcal{F}_{K45,A}), gfp(\mathcal{F}_{K45,A})\}$ which defines the K45-well-founded semantics exists.

The logic S5

McDermott proves that nonmonotonic version of S5 is same as S5.

5 Conclusion

The operator, F_P, associated with a program P that is used to define stable models, is an anti-monotonic operator that always exhibits oscillating behavior. When we draw the graph of this operator (e.g. Figures 1 and 2), these oscillations manifest themselves as cycles. The relationship between these cycles or "strict stable classes" and stable model semantics has already been exhibited in [BS90]. In this paper, we show that the well-founded semantics for logic programs can be captured within the same framework, and in fact, that the well-founded semantics corresponds precisely to a specific stable class.

We show, in addition, that the stable class that captures the well-founded semantics is minimal w.r.t. Hoare's ordering on stable classes. The stable class semantics, dually defined, is defined in terms of Smyth's ordering. Differing semantics for logic programming could be obtained by varying the ordering on stable classes, and then choosing minimal classes w.r.t. these different orderings. Hence, we feel that stable models may eventually yield a common theoretical framework based on which alternative semantics for logic programming may be developed.

An advantage of this kind of duality, and indeed other results on stable classes, is that we are now in a position to develop a well-founded semantics for default logic. Similar Hoare-Smyth dualities exist for default logic.

Previous studies of stable classes [BS90] were unable to handle auto-epistemic logics due to the fact that operators associated with such theories were not anti-monotonic (cf. Example 4.1). This situation is rectified here: an expansion class semantics for autoepistemic logic is developed using the operators of Marek and Truszczynski [MT89b]. An easy consequence of the stable class semantics is that we now have a well-founded semantics for auto-epistemic logic.

Acknowledgements

We thank Wiktor Marek for useful discussions. We greatly appreciate the financial support provided by the National Science Foundation (Nr. IRI-89-16059) and the Army Research Office (Nr. DAAL-03-88-K0087). Chitta Baral was also supported by a dissertation fellowship of the graduate school of University of Maryland.

References

[BF88] N. Bidoit and C. Froidevaux. General Logical Databases and Programs: Default logic semantics and Stratification. *Journal of Information and Computation*, 1988. (in print).

[BS90] C. Baral and V. S. Subrahmanian. Stable and Extension Class Theory for Logic Programs and Default Logics. Technical Report CS-TR-2402, Dept of Computer Science, University of Maryland, February 1990. Presented at the Third International Workshop on Non-Monotonic reasoning, South Lake Tahoe, May 90 and extended version to appear in Journal of Automated Reasoning.

[Gel89] A. Van Gelder. The alternating fixpoint of logic programs with negation. In *Proceedings of the Symposium on Principles of Database Systems*, 1989.

[GL88] M. Gelfond and V. Lifschitz. The Stable Model Semantics for Logic Programming. In R.A. Kowalski and K.A. Bowen, editors, *Proc. 5^{th} International Conference and Symposium on Logic Programming*, pages 1070–1080, Seattle, Washington, August 15-19, 1988.

[GL89] M. Gelfond and V. Lifschitz. Logic Programs with Classical Negation.

[Kon88] K. Konolige. On the relation between default and autoepistemic logic. *Artificial Intelligence*, 35:343–382, 1988.

[L87] J. Lloyd. Foundations of Logic Programming. *Springer–Verlag*, 2nd edition, 1987.

[Mar89] W. Marek. Stable theories in autoepistemic logic. *Fundamenta Informaticae*, 12:243–254, 1989.

[MS88] W. Marek and V.S. Subrahmanian. The Relationship Between Stable, Supported, Default and Auto-Epistemic Semantics for General Logic Programs. In G. Levi and M. Martelli, editors, *Proc. of ICLP 89*, pages 600-620. Extended version to appear in *Theoretical Computer Science.*

[MT89a] W. Marek and M. Truszczynski. Relating autoepistemic and default logics. Technical report, Department of Computer Science, University of Kentucky at Lexington, 1989. Also appears in Proceedings of KR 89.

[MT89b] W. Marek and M. Truszczynski. Stable semantics for logic programs and default theories. In *Proceedings of NACLP 89*, pages 243–256, 1989.

[MTS91] W. Marek, G. Shvarts and M. Truszczynski. Modal Non-Monotonic Logics: Ranges, Characterization, Computation To appear in Proceedings of KR 91.

[McD82] D. McDermott. Non-Monotonic Logic II: Non-Monotonic Modal Theories. Journal of the ACM, 29:33-57, 1982.

[Moo85] R. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25:75–94, 1985.

[Prz89] T. Przymusinski. Three-valued Formalizations of Non-monotonic Reasoning and Logic programming. In *Proceedings of First International Conference on Knowledge Representation and Reasoning*, 1989.

[Rei80] R. Reiter. A Logic for Default Reasoning. *Artificial Intelligence*, 13(1 and 2):81–132, April 1980.

[SH90] G. Shvarts. Auto-epistemic Modal Logics, In R. Parikh, ed. Proceedings of TARAK-1990, pps 97-109, Morgan Kaufmann.

[Smy78] M. Smyth. Power Domains. In *Journal of Computer and System Sciences*, volume 16,1, pages 23–36, 1978.

[VRS88] A. Van Gelder, K. Ross, and J.S. Schlipf. Unfounded Sets and Well-founded Semantics for General Logic Programs. In *Proc. 7^{th} Symposium on Principles of Database Systems*, pages 221–230, 1988. to appear in JACM.