A topological characterization of the stable and minimal model classes of propositional logic programs

Audrey P. Ferry
University of Michigan, USA

In terms of the arithmetic hierarchy, the complexity of the set of minimal models and of the set of stable models of a propositional general logic program has previously been described. However, not every set of interpretations of this level of complexity is obtained as such a set. In this paper we identify the sets of interpretations which are minimal or stable model classes by their properties in an appropriate topology on the space of interpretations. Closely connected with the topological characterization, in parallel with results previously known for stable model classes we obtain for minimal model classes both a normal-form representation as the set of minimal models of a prerequisite-free program and a logical description in terms of formulas. Our approach centers on the relation which we establish between stable and minimal model classes. We include examples of calculations which can be performed by these methods.

1. Introduction

Let P be a propositional general logic program over a finite or countably infinite set U of atoms. Recall that an interpretation for P is a subset of U. By a model of P we mean an interpretation $S \subseteq U$ which is closed under T_P; by a minimal model of P we mean a model S of P such that no proper subset $S' \subsetneq S$ is a model of P; and by a stable model of P we mean a set $S \subseteq U$ such that $S =$ least fixpoint of $T_{GL_S(P)}$, where $GL_S(P)$ is the Gelfond–Lifschitz transform of program P with respect to S defined in [3]. Let $Mod(P)$, $Min(P)$, and $Stab(P)$ denote, respectively, the set of models, minimal models, and stable models of P. For a given program P, we see that these classes are related by

$$Stab(P) \subseteq Min(P) \subseteq Mod(P) \subseteq 2^U,$$

with the first inclusion given by [3].

We refer to a subset $\mathcal{S}$ of 2^U as a model class, minimal model class, or stable model class if $\mathcal{S} = Mod(P)$, $\mathcal{S} = Min(P)$, or $\mathcal{S} = Stab(P)$, respectively, for some program P. In this paper we are concerned with discovering which subsets of 2^U arise as minimal and stable model classes. We find simple mathematical properties that characterize these classes.

© J.C. Baltzer AG, Science Publishers

In [4], the important notion of "proof scheme" was introduced. Informally, a proof scheme from P for element $u \in U$ consists of: (1) a finite sequence of clauses from P, with u as the head of the last clause, such that every positive literal in the body of a clause occurs as the head of some previous clause, and (2) a cumulative list of the negative literals occurring in these clauses, called the "support" of the proof scheme. Using this notion, it was shown that the stable models S of a program P could be described by a system of equivalences of the form

$$u \in S \leftrightarrow \bigvee_{A \in \Psi_u} \bigwedge_{a \in A} a \notin S$$

for $u \in U$, where the sets $A \in \Psi_u$ are the supports of minimal proof schemes for u. Note that each conjunction in this expression is finite, but that the disjunction may be finite or countably infinite. In [6] these conditions are called a system of "defining equations" for the stable models of P, since the equivalences may be considered as Boolean equations which are satisfied by those valuations which correspond to stable models. Alternately these conditions may be formulated as a sentence $\sigma =$

$$\bigwedge_{u \in U} [R(\underline{u}) \leftrightarrow \bigvee_{A \in \Psi_u} \bigwedge_{a \in A} \neg R(\underline{a})]$$

of $\mathcal{L}^U_{\omega_1,\omega}$ (in the language $\mathcal{L}^U$ consisting of a unary predicate symbol R and a constant symbol $\underline{u}$ for each $u \in U$), with the property that $(U, S, \vec{u}_{u \in U}) \models \sigma$ if and only if S is a stable model.

From this description, it was shown in [4] that the class of stable models of a program P is Π^0_2 in the parameter P. In the same paper, it was shown, by a completely different method, that the class of minimal models of P is also Π^0_2 in P. The present author then found an example of a Π^0_2 antichain which cannot be realized either as the class of stable models or as the class of minimal models of any program. To characterize stable and minimal model classes and to distinguish between them, a narrower description of the complexity of each is needed.

One such description is given in the paper [5]. There it is shown that the sets of Turing degrees realized in stable model classes of recursive programs correspond exactly to the sets of degrees realized in Π^0_1 subsets of U^U. In this paper, on the other hand, we characterize stable model classes not up to Turing degree, but in absolute terms. The description which is provided here measures complexity however not in the arithmetic hierarchy but rather in the Borel hierarchy of a suitable topology on 2^U. Though the results could be translated into recursion–theoretic language, the topological language is more direct.

We recall first the definition of the Cantor topology on 2^U, which is familiar from descriptive set theory and other foundational studies. In this space the open sets are generated by basis elements of the form $\{S : \bigwedge_{a \in A} a \in S \wedge \bigwedge_{b \in B} b \notin S\}$ for A and B finite, possibly empty, subsets of U. This is of course the product topology on 2^U for $2 = \{0, 1\}$ under the discrete topology. It is both metrizable and compact. Its Borel and analytic hierarchies are analogous and related to the arithmetic and analytical hierarchies of recursion theory.

As was shown in [4], the model classes of logic programs are easily characterized as the closed sets of 2^U under the Cantor topology (see Theorem 1.1 below). The result that *Stab*(P) is Π_2^0 in P mentioned above means that every stable model class is a G_δ subset of 2^U, and similarly for minimal model classes. Examples 1 and 9 below, which are adaptations to the topological setting of the author's earlier example, show that neither stable nor minimal model classes are characterized by the properties of being an antichain and a G_δ set in this topology.

An exact description of these classes can be given, however, in terms of a different, coarser topology on 2^U, which, unlike the Cantor topology or notions from recursion theory, is not symmetric with respect to positive and negative information. We take as a basis for this topology all sets of the form $\{S : \bigwedge_{b\in B} b \notin S\}$ for B a finite, again possibly empty, subset of U. This turns out to be an instance of the inverse–Scott topology encountered in the study of continuous lattices, with 2^U regarded as a lattice under the partial ordering $\subseteq$. The same topology has been investigated in connection with logic programming by Aida Batarekh in her thesis [1] and subsequent papers.

The essential feature of this topology is that open sets are closed under "going down" (that is, if $\mathcal{G}$ is open, $S \in \mathcal{G}$, and $S' \subseteq S$, then $S' \in \mathcal{G}$) and closed sets under "going up" (if $\mathcal{F}$ is closed, $S \in \mathcal{F}$, and $S' \supseteq S$, then $S' \in \mathcal{F}$). We note that the only open set to which U belongs is $\{S : \bigwedge_{b\in\emptyset} b \notin S\} = 2^U$; equivalently, U belongs to every closed set except $\emptyset$. Similarly the empty subset of U is an element of every open set except $\emptyset$; the only closed set to which it belongs is 2^U itself. Note also that $\{\emptyset\}$, though closed under $\subseteq$, is not open.

In contrast to the Cantor space, much about the inverse–Scott space runs counter to intuition. For one thing, its Borel hierarchy is not analogous to the arithmetic hierarchy; a closed set, for instance, generally is not G_δ. The situation is much worse, however. The space is not Hausdorff—in fact does not even satisfy the weaker requirement that singleton sets be closed—so convergence does not work as expected. For instance, an increasing sequence $S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots$ in 2^U converges not only to the point $\bigcup_{i\in\omega} S_i$, but also to every point S' such that $\bigcup_{i\in\omega} S_i \subseteq S'$. Nonetheless the topological representation is natural, and its peculiarities seem to reflect real difficulties in the logical behavior we are trying to describe.

We make use of the following notation: The spaces described above will be denoted by $(2^U, \mathit{Cantor})$ and $(2^U, \mathit{Scott}^{-1})$. For $\mathcal{S} \subseteq 2^U$, we let $\bar{\mathcal{S}}$ represent the closure of $\mathcal{S}$ in the Cantor topology, and $\tilde{\mathcal{S}}$ its closure in the inverse–Scott topology. We use *min* $\mathcal{S}$ for the set of inclusion–minimal elements of $\mathcal{S}$, and $\mathcal{S}^{\supseteq}$ for the collection of supersets of elements of $\mathcal{S}$ (that is, $\{S' : S' \supseteq S$ for some $S \in \mathcal{S}\}$). For B a finite subset of U, the symbol $\vec{B}$ represents the sequence $\langle b_1, \ldots, b_n\rangle$ if $B = \{b_1, \ldots, b_n\}$, or the empty sequence if $B = \emptyset$; likewise $\neg\vec{B}$ represents $\langle \neg b_1, \ldots, \neg b_n\rangle$ or the empty sequence. An expression of the form

$\vec{A}^1, \dots, \vec{A}^n, \neg\vec{B}^1, \dots, \neg\vec{B}^m$ represents a concatenation of finite sequences. We sometimes write a general program clause in the form $c \leftarrow \vec{A}^1, \dots, \vec{A}^n, \neg\vec{B}^1, \dots, \neg\vec{B}^m$ for $A^1, \dots, A^n, B^1, \dots, B^n$ finite subsets of U, permitting the possibility that any or all of the sets $A^1, \dots, A^n, B^1, \dots, B^m$ might be empty. The unnegated atoms in the body of a clause are called prerequisites, and if the set of prerequisites is empty, then the clause is called prerequisite–free. A program is called prerequisite–free if it consists entirely of prerequisite–free clauses. We follow the convention that $\bigwedge_{a\in\emptyset} \varphi_a = \top$ and $\bigvee_{a\in\emptyset} \varphi_a = \bot$. Finally we permit the use of infinitary conjunction or alternative symbols outside the setting of a formal system as abbreviations for saying that all or some of an indicated collection of conditions hold. We use the expression $\vec{B} \notin S$ (for $S \subseteq U$) to stand for $\bigwedge_{b\in B} b \notin S$.

For our purposes, the following result is fundamental:

Theorem 1.1. (a) [4] *A set $\mathcal{S} \subseteq 2^U$ is the model class of some program if and only if $U \in \mathcal{S}$ and $\mathcal{S}$ is closed in the Cantor topology.*
(b) *A set $\mathcal{S} \subseteq 2^U$ is the model class of some prerequisite–free program if and only if $\mathcal{S}$ is nonempty and closed in the inverse–Scott topology.*

Proof
(a) $\Rightarrow$ Let P be a program such that $\mathcal{S} = Mod(P)$. Then

$$\begin{aligned} \mathcal{S} &= \{S : \text{for every clause } \rho \text{ in } P,\ S \models \rho\} \\ &= \bigcap_{\rho\in P} \{S : S \models \rho\}. \end{aligned}$$

So it suffices to show that, for any clause ρ, $U \models \rho$ and $\{S : S \models \rho\}$ is closed. Suppose that ρ has the form

$$c \leftarrow \vec{A}, \neg\vec{B}$$

for A, B finite, possibly empty, subsets of U. Then

$$\begin{aligned} \{S : S \models \rho\} &= \{S : S \models (\bigwedge_{a\in A} a \wedge \bigwedge_{b\in B} \neg b) \rightarrow c\} \\ &= \{S : S \models \bigvee_{a\in A} \neg a \vee \bigvee_{b\in B} b \ \vee c\} \\ &= \{S : \bigvee_{a\in A} a \notin S \vee \bigvee_{b\in B} b \in S \ \vee \ c \in S\}, \end{aligned}$$

which is basic closed in the Cantor topology. Since $c \in U$, $U \models \rho$.

$\Leftarrow$

Suppose that $\mathcal{S} = \bigcap_{i \in I} \mathcal{S}_i$, where, for each i, $\mathcal{S}_i$ is basic closed and $U \in \mathcal{S}_i$. We will show that for each i there is a program clause ρ_i such that $\mathcal{S}_i = \{S : S \models \rho_i\}$. So suppose that the basic closed set $\mathcal{S}_i$ has the form

$$\{S : \bigvee_{a \in A} a \notin S \vee \bigvee_{b \in B} b \in S\}$$

for finite subsets A and B of U. Since $U \in \mathcal{S}_i$, B must be nonempty. Let c be any element of B. Then

$$\begin{aligned}\mathcal{S}_i &= \{S : \bigvee_{a \in A} a \notin S \vee \bigvee_{b \in B - \{c\}} b \in S \ \vee\ c \in S\} \\ &= \{S : S \models \bigvee_{a \in A} \neg a \vee \bigvee_{b \in B - \{c\}} b \vee c\} \\ &= \{S : S \models (\bigwedge_{a \in A} a \wedge \bigwedge_{b \in B - \{c\}} \neg b) \to c\} \\ &= \{S : S \models \rho_i\},\end{aligned}$$

where ρ_i is the clause $c \leftarrow \vec{A}, \neg\vec{B}$. Finally let P be the program consisting of $\{\rho_i : i \in I\}$. Then $\mathcal{S} = Mod(P)$.

(b) The proof of part (b) is similar. ■

In the remainder of the paper, we extend the perspective of Theorem 1.1 to the minimal and stable model classes of programs. Guided by the topological interpretation, we will show that for every program P there is a prerequisite–free program P' with the same minimal models. For programs in this form we are able both to characterize their minimal model classes topologically and to describe clearly which sets are minimal models in terms of the program clauses. From this description we can then explain what additional property distinguishes the stable models among the minimal models and, further, describe the relation of stable model classes to minimal model classes. Specifically, we obtain for minimal model classes a prerequisite–free representation and defining equations similar to those for stable model classes, and, for both minimal and stable model classes, a characterization theorem.

2. Minimal model classes

We begin with general facts about subsets of $(2^U, Cantor)$ and $(2^U, Scott^{-1})$ and their minimal elements:

Proposition 2.1. *For nonempty* $\mathcal{S} \subseteq 2^U$:
(a) *min* $\mathcal{S} =$ *min* $\mathcal{S}^{\supseteq}$. *In fact, min* $\mathcal{S} =$ *min* $\mathcal{X}$ *for any* $\mathcal{X}$ *with* $\mathcal{S} \subseteq \mathcal{X} \subseteq \mathcal{S}^{\supseteq}$.
(b) $\mathcal{S} =$ *min* $\mathcal{S}$ *if and only if* $\mathcal{S}$ *is an antichain.*

Proof. Straightforward. ∎

Proposition 2.2. *For nonempty* $\mathcal{S} \subseteq 2^U$:
(a) $\mathcal{S}^{\supseteq} \subseteq \tilde{\mathcal{S}}$.
(b) $\mathcal{S}^{\supseteq}$ *is closed in the inverse–Scott topology iff* $\mathcal{S}^{\supseteq} = \tilde{\mathcal{S}}$.

Proof
(a) As noted above, closed sets in $(2^U, Scott^{-1})$ are closed under the relation $\supseteq$. Since $\tilde{\mathcal{S}}$ is closed and contains $\mathcal{S}$, it contains $\mathcal{S}^{\supseteq}$ as well.
(b) If $\mathcal{S}^{\supseteq}$ is closed, then $\tilde{\mathcal{S}} \subseteq \mathcal{S}^{\supseteq}$ by definition of closure. ∎

Proposition 2.3. *If* $\mathcal{F} \subseteq 2^U$ *is closed in the Cantor topology, then* $\mathcal{F}^{\supseteq}$ *is closed in the inverse–Scott topology.*

Proof. Suppose that $\mathcal{F}^{\supseteq}$ is not closed in $(2^U, Scott^{-1})$. Let $R \in \widetilde{\mathcal{F}^{\supseteq}} - \mathcal{F}^{\supseteq}$. Because $R \in \widetilde{\mathcal{F}^{\supseteq}}$, there is a sequence $S_1, S_2, \ldots$ of elements of $\mathcal{F}^{\supseteq}$ converging in the inverse–Scott sense to R. (This requires only that the space be first countable—that is, that each point have a countable neighborhood basis.)

By definition of $\mathcal{F}^{\supseteq}$, each S_i has a subset S_i' with $S_i' \in \mathcal{F}$. Since $(2^U, Cantor)$ is compact and $\mathcal{F}$ is a closed subset, $\mathcal{F}$ itself is compact. Therefore the sequence $S_1', S_2', \ldots$ in $\mathcal{F}$ has a subsequence which converges in the Cantor sense to a point $S^* \in \mathcal{F}$.

We know that $S^* \nsubseteq R$ because $R \notin \mathcal{F}^{\supseteq}$. So there is a point $x \in U$ such that $x \in S^*$ and $x \notin R$. On the one hand, because $S_i \to R$ in $(2^U, Scott^{-1})$ and $x \notin R$, there is a number N such that, for all $i > N$, $x \notin S_i$. Then $x \notin S_i'$ as well, because $S_i' \subseteq S_i$. On the other hand, we have that $x \in S_i'$ for infinitely many $i > N$, because some subsequence of S_i' converges in $(2^U, Cantor)$ to S^* and $x \in S^*$. We conclude by contradiction that $\mathcal{F}^{\supseteq}$ is inverse–Scott closed. ∎

Though they are not needed directly, we find the following observations useful for understanding how the topologies are related:

Corollary 2.1
(a) *For* $\mathcal{S} \neq \emptyset$, $\tilde{\mathcal{S}} = (\bar{\mathcal{S}})^{\supseteq}$.
(b) $\mathcal{S}^{\supseteq}$ *is closed in the Cantor sense if and only if it is closed in the inverse–Scott sense.*

Proof

(a) We note that $\mathcal{S} \subseteq (\bar{\mathcal{S}})^{\supseteq}$. By Proposition 2.3, $(\bar{\mathcal{S}})^{\supseteq}$ is inverse–Scott closed. So by the definition of closure, $\tilde{\mathcal{S}} \subseteq (\bar{\mathcal{S}})^{\supseteq}$. Similarly, since $\mathcal{S} \subseteq \tilde{\mathcal{S}}$ and $\tilde{\mathcal{S}}$ is closed in the Cantor topology, $\bar{\mathcal{S}} \subseteq \tilde{\mathcal{S}}$. Hence $(\bar{\mathcal{S}})^{\supseteq} \subseteq (\tilde{\mathcal{S}})^{\supseteq}$. By Proposition 2.2(a), $(\tilde{\mathcal{S}})^{\supseteq} \subseteq \tilde{\mathcal{S}}$, so $(\bar{\mathcal{S}})^{\supseteq} \subseteq \tilde{\mathcal{S}}$.

(b) The "if" direction holds because the Cantor topology extends the inverse–Scott topology. The "only if" direction uses Proposition 2.3. ■

Lemma 2.1. *If $\mathcal{F} \subseteq 2^U$ is closed in the Cantor topology, then for all $X \in \mathcal{F}$ there exists $X' \subseteq X$ such that X' is a minimal element of $\mathcal{F}$.*

Proof. If $\mathcal{F} = \emptyset$, then the conclusion is vacuously true. So assume $\mathcal{F} \neq \emptyset$, and let $X \in \mathcal{F}$. We apply Zorn's Lemma to the set $P = \{X' \in \mathcal{F} : X' \subseteq X\}$ with order relation $<$ given by $\supsetneq$.

Let $X_1 \supsetneq X_2 \supsetneq \dots$ be a chain in P. Let $Y = \bigcap X_i$. Clearly $Y \subsetneq X_i$ for each i, and $Y \subseteq X$. Since $\mathcal{F}$ is closed, we also have that $Y \in \mathcal{F}$. So Y is an upper bound (in the sense of $<$) for the given chain. By Zorn's Lemma, P has a $<$-maximal element X'.

X' is then $\subseteq$-minimal with respect to P. But if $S \subseteq X'$ for any $S \in \mathcal{F}$, then $S \in P$, so X' is a minimal element of $\mathcal{F}$. ■

Theorem 2.1 (Characterization of minimal model classes). *For nonempty $\mathcal{S} \subseteq 2^U$, the following are equivalent:*

(i) *$\mathcal{S}$ is a minimal class.*

(ii) *$\mathcal{S}$ is an antichain and $\mathcal{S}^{\supseteq}$ is inverse–Scott closed.*

(iii) $\mathcal{S} = \min \tilde{\mathcal{S}}$.

(iv) $\mathcal{S} = \min \bar{\mathcal{S}}$.

Proof

(i) ⇒ (ii)

Let P be a program such that $\mathcal{S} = Min(P) = \min Mod(P)$, and let $\mathcal{F} = Mod(P)$. By Theorem 1.1, $\mathcal{F}$ is closed in the Cantor topology. By Proposition 2.3, $\mathcal{F}^{\supseteq}$ then is closed in the inverse–Scott topology. We show that $\mathcal{S}^{\supseteq} = \mathcal{F}^{\supseteq}$:

Since $\mathcal{S} \subseteq \mathcal{F}$, $\mathcal{S}^{\supseteq} \subseteq \mathcal{F}^{\supseteq}$. For the other inclusion, the preceding lemma says that $\mathcal{F} \subseteq \mathcal{S}^{\supseteq}$, which then implies $\mathcal{F}^{\supseteq} \subseteq \mathcal{S}^{\supseteq}$.

Therefore $\mathcal{S}^{\supseteq}$ is an inverse–Scott closed set.

That $\mathcal{S}$ is an antichain was shown in [4]; for completeness, we include a proof here as well. Since $\mathcal{S} \subseteq \mathcal{F}$ and $\mathcal{F} \subseteq \mathcal{S}^{\supseteq}$ (by the lemma), Proposition 2.1(a) gives $\min \mathcal{S} = \min \mathcal{F}$. By definition, $\min \mathcal{F} = \mathcal{S}$. Since then $\min \mathcal{S} = \mathcal{S}$, $\mathcal{S}$ is an antichain by Proposition 2.1(b).

(ii) ⇒ (iii)

Since $\mathcal{S}^{\supseteq}$ is closed, $\tilde{\mathcal{S}} = \mathcal{S}^{\supseteq}$ by Proposition 2.2(b). Then $\min \tilde{\mathcal{S}} = \min \mathcal{S}^{\supseteq} = \min \mathcal{S} = \mathcal{S}$ by Proposition 2.1(a) and (b).

(iii) ⇒ (iv)
$\bar{\mathcal{S}} \subseteq \tilde{\mathcal{S}}$ because $\tilde{\mathcal{S}}$ is closed in the Cantor sense. By Proposition 2.3, $(\bar{\mathcal{S}})^{\supseteq}$ is closed in the inverse–Scott sense, so $\tilde{\mathcal{S}} \subseteq (\bar{\mathcal{S}})^{\supseteq}$. Since $\tilde{\mathcal{S}}$ is sandwiched between $\bar{\mathcal{S}}$ and $(\bar{\mathcal{S}})^{\supseteq}$, by Proposition 2.1(a) $\min \bar{\mathcal{S}} = \min \tilde{\mathcal{S}}$.

(iv) ⇒ (i)
By Theorem 1.1, $\bar{\mathcal{S}} \cup \{U\}$ is the collection of models of some program P. Then $\mathcal{S} = \min \bar{\mathcal{S}} = \min(\bar{\mathcal{S}} \cup \{U\}) = \min \mathit{Mod}(P) = \mathit{Min}(P)$. ■

By way of this theorem, we can now present an example to show that the G_δ and antichain properties mentioned in the introduction are too broad to characterize minimal model classes:

Example 2.1. Let $\mathcal{S} = \{\{u_0\}, \{u_1\}, \{u_2\}, \ldots\}$, where $U = \{u_0, u_1, u_2, \ldots\}$. We note that $\mathcal{S}$ is both an antichain and a G_δ subset in $(2^U, \mathit{Cantor})$. But condition (iv) of the theorem, namely $\mathcal{S} = \min \bar{\mathcal{S}}$, fails because $\emptyset \in \bar{\mathcal{S}}$, and therefore $\mathcal{S}$ is not a minimal model class. Alternately, we can check, as in condition (ii), that

$$\mathcal{S}^{\supseteq} = \{S : u_0 \in S \vee u_1 \in S \vee u_2 \in S \vee \ldots\} = 2^U - \{\emptyset\},$$

which, as observed in the Introduction, is not closed in $(2^U, \mathit{Scott}^{-1})$. ■

From the propositions preceding the theorem, we also get a useful representation result for minimal model classes:

Theorem 2.2 (Prerequisite–free representation). *For any program P there is a prerequisite–free program P' which has the same minimal models as P.*

Proof. By Theorem 1.1(a), $\mathit{Mod}(P)$ is a nonempty closed subset $\mathcal{F}$ of $(2^U, \mathit{Cantor})$. By Proposition 2.3, $\mathcal{F}^{\supseteq}$ is closed in $(2^U, \mathit{Scott}^{-1})$. By Theorem 1.1(b) then $\mathcal{F}^{\supseteq} = \mathit{Mod}(P')$ for some prerequisite–free P'. Proposition 2.1(a) says that $\mathcal{F}$ and $\mathcal{F}^{\supseteq}$ have the same minimal elements. Thus $\mathit{Min}(P) = \min \mathit{Mod}(P) = \min \mathcal{F} = \min \mathcal{F}^{\supseteq} = \min \mathit{Mod}(P') = \mathit{Min}(P')$. ■

To study minimal model classes, then, it is enough to study the minimal models of programs which are prerequisite–free. According to Theorem 1.1(b), this amounts to studying the minimal elements of sets which are inverse–Scott closed. We show next how to describe these minimal elements in terms of the closed–set structure.

Proposition 2.4. *Let $\mathcal{F} = \bigcap_{C \in \mathcal{C}} C$ be a nonempty closed subset of 2^U in the inverse–Scott topology, where each $C \in \mathcal{C}$ is a basic closed set of the form*

$C = \{S : \bigvee_{y \in P_C} y \in S\}$ for P_C a finite, nonempty subset of U. (We permit $\mathcal{C} = \emptyset$ in order to include $\mathcal{F} = 2^U$ as a possibility.) Let

$$\mathcal{M} = \{S : \bigwedge_{u \in U} [u \in S \leftrightarrow \bigvee_{\substack{C \in \mathcal{C},\\ u \in P_C}} \bigwedge_{\substack{y \in P_C,\\ y \neq u}} y \notin S]\}.$$

Then $\mathcal{M}$ is the set of minimal elements of $\mathcal{F}$.

Proof
[To show: $\mathcal{M} \subseteq \min \mathcal{F}$]
Let $S' \in \mathcal{M}$.

Case 1. Suppose first that $S' = \emptyset$. For all the equivalences above to hold in this case, it must be that $\mathcal{C} = \emptyset$, which means that $\mathcal{F} = 2^U$. Hence $S' \in \mathcal{F}$. But since S' has no proper subsets, $S' \in \min \mathcal{F}$.

Case 2. So suppose that $S' \neq \emptyset$.
First we check that $S' \in \mathcal{F}$. It suffices to show that $S' \in C$ for each $C \in \mathcal{C}$. Suppose that

$$C = \{S : \bigvee_{y \in P_C} y \in S\}.$$

Let $u' \in P_C$.
If $u' \in S'$, then $S' \in C$ by definition of C.
If $u' \notin S'$, then, because

$$u' \in S' \leftrightarrow \bigvee_{\substack{C \in \mathcal{C},\\ u' \in P_C}} \bigwedge_{\substack{y \in P_C,\\ y \neq u'}} y \notin S',$$

we have that $y' \in S'$ for some $y' \in P_C$ with $y' \neq u'$, so $S' \in C$ in this case as well.
Now we show that no proper subset of S' belongs to $\mathcal{F}$. Recalling that $S' \neq \emptyset$, let $A \subsetneq S'$. Choose $a \in U$ such that $a \in S'$ and $a \notin A$. Since

$$a \in S' \leftrightarrow \bigvee_{\substack{C \in \mathcal{C},\\ a \in P_C}} \bigwedge_{\substack{y \in P_C,\\ y \neq a}} y \notin S',$$

there is some $C \in \mathcal{C}$ with $a \in P_C$ such that $b \notin S'$ for all $b \in P_C$ with $b \neq a$. Since $A \subseteq S'$, $b \notin A$ as well, for each such b. Since a was chosen so that $a \notin A$, no element of P_C belongs to A. Thus $A \notin C$, and hence $A \notin \mathcal{F}$.

[To show: $\min \mathcal{F} \subseteq \mathcal{M}$]
Suppose that $S' \notin \mathcal{M}$. Then there is some $a \in U$ such that

$$[a \in S' \wedge \bigwedge_{\substack{C \in \mathcal{C},\\ a \in P_C}} \bigvee_{\substack{y \in P_C,\\ y \neq a}} y \in S'] \vee [a \notin S' \wedge \bigvee_{\substack{C \in \mathcal{C},\\ a \in P_C}} \bigwedge_{\substack{y \in P_C,\\ y \neq a}} y \notin S'].$$

Case 1. If $S' = \emptyset$, then, since the second disjunct must hold, $\mathcal{C} \neq \emptyset$. Hence $\mathcal{F} \neq 2^U$. But 2^U is the only closed set to which $\emptyset$ belongs. So $S' \notin \mathcal{F}$.

Case 2. Suppose then that $S' \neq \emptyset$. If $S' \notin \mathcal{F}$, we are done. So assume $S' \in \mathcal{F}$. Then the second disjunct is false, so the first disjunct holds. We claim that the set $S' - \{a\} \in \mathcal{F}$:
For sets $C \in \mathcal{C}$ with $a \notin P_C$, $S' \in \mathcal{F}$ requires that $b \in S'$ for some $b \in P_C$. Since $b \neq a$, b actually belongs to $S' - \{a\}$, which means that $S' - \{a\} \in C$.
For sets $C \in \mathcal{C}$ with $a \in P_C$, the first disjunct above requires that $b \in S'$ for some $b \in P_C$ with $b \neq a$. In this case as well, then, $b \in S' - \{a\}$, so $S' - \{a\} \in C$.
We conclude that $S' - \{a\} \in \bigcap_{C \in \mathcal{C}} C = \mathcal{F}$. Hence S' is not minimal in $\mathcal{F}$. ■

We can now apply this result to get a useful description of the minimal models of a logic program in terms of a system of "defining equations:"

Theorem 2.3 (Defining equations for minimal models). *Let P be a general logic program over U. For each $u \in U$, there is a countable (possibly empty) collection Φ_u of finite (possibly empty) sequences A from U, such that*

$$Min(P) = \{S : \bigwedge_{u \in U} [u \in S \leftrightarrow \bigvee_{A \in \Phi_u} \bigwedge_{a \in A} a \notin S]\}.$$

Proof. Let P' be a prerequisite-free program with the same minimal models as P. Fix an expression for $Mod(P')$ as an intersection of basic closed sets in $(2^U, Scott^{-1})$ corresponding to the clauses of P'. Then the system of equivalences formed as in the preceding proposition characterizes the minimal elements of $Mod(P')$. Following the notation of the preceding proposition, for each $u \in U$, let Φ_u consist of those sets $P_C - \{u\}$ for which $u \in P_C$. Then

$$Min(P) = Min(P') = \min Mod(P')$$

$$= \{S : \bigwedge_{u \in U} [u \in S \leftrightarrow \bigvee_{A \in \Phi_u} \bigwedge_{a \in A} a \notin S]\}.$$

■

Just as for the sentence σ in the case of stable models, we may formulate the defining equations for the minimal models of P as a sentence $\mu =$

$$\bigwedge_{u\in U}[R(\underline{u}) \leftrightarrow \bigvee_{A\in\Phi_u}\bigwedge_{a\in A}\neg R(\underline{a})]$$

of $\mathcal{L}^U_{\omega_1,\omega}$ (in the language $\mathcal{L}^U$ consisting of a unary predicate symbol R and a constant symbol $\underline{u}$ for each $u \in U$), with the property that $(U, S, \vec{u}_{u\in U}) \models \mu$ if and only if S is a minimal model of P. For convenience, we shall often use the symbol μ to refer to the conditions on a set S entailed by having $(U, S, \vec{u}_{u\in U}) \models \mu$. As with stable models, the role of negative information in specifying minimal models is made particularly apparent by the defining equations. By analogy with the stable model case, where the elements of Ψ_u are the supports of proof schemes in P for u, we refer to the elements of Φ_u as "supports" of u in the clauses of the associated program P'.

We illustrate the notion of supports and the formation of the sentence μ using a prerequisite–free program of the simplest sort as an example:

Example 2.2. Let $U = \{a, b, c, d, e\}$ and let P be the program

$$a \leftarrow \neg b$$
$$b \leftarrow \neg c$$
$$c \leftarrow \neg d$$
$$d \leftarrow \neg e.$$

As in Theorem 1.1, we express $Mod(P)$ as an intersection of sets determined by the clauses of P, and recall, because P is prerequisite–free, that each is basic closed in $(2^U, Scott^{-1})$:

$$Mod(\mathrm{P}) = \mathcal{F}_1 \cap \mathcal{F}_2 \cap \mathcal{F}_3 \cap \mathcal{F}_4, \text{ where}$$
$$\mathcal{F}_1 = \{S : a \in S \vee b \in S\}$$
$$\mathcal{F}_2 = \{S : b \in S \vee c \in S\}$$
$$\mathcal{F}_3 = \{S : c \in S \vee d \in S\}$$
$$\mathcal{F}_4 = \{S : d \in S \vee e \in S\}.$$

For the element $a \in U$, which occurs only in the first clause, the support of a in $\mathcal{F}_1$ is $\{b\}$, so the defining equation for a is

$$a \in S \leftrightarrow b \notin S.$$

For $b \in U$, the support of b in $\mathcal{F}_1$ is $\{a\}$ and the support of b in $\mathcal{F}_2$ is $\{c\}$, so the defining equation for b is

$$b \in S \leftrightarrow a \notin S \vee c \notin S.$$

The defining equations for c, d, and e are formed similarly, so the condition determining minimal models S is

$$\begin{array}{rl} & (a \in S \leftrightarrow b \notin S) \\ \wedge & (b \in S \leftrightarrow a \notin S \vee c \notin S) \\ \wedge & (c \in S \leftrightarrow b \notin S \vee d \notin S) \\ \wedge & (d \in S \leftrightarrow c \notin S \vee e \notin S) \\ \wedge & (e \in S \leftrightarrow d \notin S). \end{array}$$

Formally the sentence μ in the language $\mathcal{L}^U$ has the form

$$\begin{array}{rl} & [R(\underline{a}) \leftrightarrow \neg R(\underline{b})] \\ \wedge & [R(\underline{b}) \leftrightarrow \neg R(\underline{a}) \vee \neg R(\underline{c})] \\ \wedge & [R(\underline{c}) \leftrightarrow \neg R(\underline{b}) \vee \neg R(\underline{d})] \\ \wedge & [R(\underline{d}) \leftrightarrow \neg R(\underline{c}) \vee \neg R(\underline{e})] \\ \wedge & [R(\underline{e}) \leftrightarrow \neg R(\underline{d})], \end{array}$$

but we will generally bypass the formal statement and display as μ the preceding conjunction of equivalences. Thus the minimal models of P are

$$\mathcal{M} = \{\{a, c, d\}, \{a, c, e\}, \{b, c, e\}, \{b, d\}\}. \qquad \blacksquare$$

Next we present a pair of algorithms which will allow us to find the sentence μ and the minimal models for more complicated programs. This will also give us the means to find a prerequisite–free program with the same minimal models as a given program, or indeed to find a program with some specified set $\mathcal{M}$ as its minimal models, as long as $\mathcal{M}$ satisfies the conditions, as in Theorem 2.1, of a minimal model class.

We will need the following lemma:

Lemma 2.2. *Let $\mathcal{F} \subseteq 2^U$. Suppose that $\mathcal{F}^{\supseteq}$ is expressed in the form*

$$\mathcal{F}^{\supseteq} = \bigcup_{X \in \mathcal{X}} \{S : \bigwedge_{x \in X} x \in S\} = \bigcup_{X \in \mathcal{X}} \{S : X \subseteq S\}$$

where $\mathcal{X} \subseteq 2^U$ is an antichain. Then

$$\min \mathcal{F} = \{X : X \in \mathcal{X}\} = \mathcal{X}.$$

Proof

($\supseteq$) Let $Y \in \mathcal{X}$. By Proposition 2.1, $\min \mathcal{F} = \min \mathcal{F}^{\supseteq}$, so it will suffice to show that $Y \in \min \mathcal{F}^{\supseteq}$. First $Y \in \mathcal{F}^{\supseteq}$ because $Y \in \{S : Y \subseteq S\}$. Now suppose that $Z \in \mathcal{F}^{\supseteq}$ and $Z \subseteq Y$. Then $X \subseteq Z$ for some $X \in \mathcal{X}$. But since $X \subseteq Z \subseteq Y$ and $\mathcal{X}$ is an antichain, $X = Y$. Hence $Y \subseteq Z$, which shows that Y is minimal in $\mathcal{F}^{\supseteq}$.

($\subseteq$) Let $Y \in \min \mathcal{F}$. By Proposition 2.1, $Y \in \min \mathcal{F}^{\supseteq}$. Since $Y \in \mathcal{F}^{\supseteq}$, there is some set $X \in \mathcal{X}$ such that $X \subseteq Y$. But because Y is minimal in $\mathcal{F}^{\supseteq}$ and $X \in \mathcal{F}^{\supseteq}$, $X = Y$. Hence $Y \in \mathcal{X}$. ■

The first algorithm contains a method for computing $\mathcal{F}^{\supseteq}$ from a closed set $\mathcal{F}$ in the Cantor topology. From the expression which is arrived at for $\mathcal{F}^{\supseteq}$, the minimal elements of $\mathcal{F}$ can be read off directly.

Algorithm 2.1 (for obtaining Min(*P*) from *P*)

Let P be a program, not necessarily prerequisite–free. Fix an expression for the closed set $\mathcal{F} = Mod(P)$ in $(2^U, Cantor)$ as an intersection of basic closed sets determined by the clauses of P. $\mathcal{F}$ appears then as an intersection of sets defined by finite disjunctions.

(1) If P consists of a single clause, express $\mathcal{F}$ as a union of sets of the form $\{S : u \in S\}$ or $\{S : u \notin S\}$ corresponding to the disjuncts of the clause. If every clause in P has an empty body, then express $\mathcal{F}$ as a single set defined by a conjunction (possibly infinite) of the expressions "$u \in S$" for clauses $u \leftarrow$ in P. Otherwise, use the distributive property to express $\mathcal{F}$ as a union of conjunctively defined sets.

(2) Simplify by omitting any summand which is empty.

(3) Since $\supseteq$ commutes with $\cup$, form $\mathcal{F}^{\supseteq}$ by applying $\supseteq$ to each summand.

(4) For each summand, evaluate the application of $\supseteq$ by omitting the negative conjuncts.

(5) Simplify by retaining only maximal summands. That is, if $\mathcal{S}_1$ and $\mathcal{S}_2$ are summands, and the conjunction defining $\mathcal{S}_1$ is a proper subexpression of the conjunction defining $\mathcal{S}_2$, omit $\mathcal{S}_2$.

Then $\mathcal{F}^{\supseteq}$ is expressed as a union of sets of the form

$$\{S : \bigwedge_{u \in X} u \in S\} = \{S : X \subseteq S\},$$

where the sets X determining each summand form an antichain. Hence, by the preceding lemma, the sets X themselves are the minimal models of P. ■

Before giving an example, we present the second algorithm, which will be used to find prerequisite–free programs under appropriate conditions. Again we require a lemma:

Lemma 2.3. *Let $\mathcal{X} \subseteq 2^U$ be given. Let*

$$\mathcal{Z} = \{Z \subseteq U : Z \text{ is finite and for all } \mathrm{X} \in \mathcal{X}, \mathrm{Z} \cap \mathrm{X} \neq \emptyset\}.$$

Then

$$\widetilde{\mathcal{X}^{\supseteq}} = \bigcap_{Z \in \mathcal{Z}} \{S : Z \cap S \neq \emptyset\}.$$

Proof.

($\subseteq$) For each $Z \in \mathcal{Z}$,

$$\{S : Z \cap S \neq \emptyset\} = \{S : \bigvee_{u \in Z} u \in S\}$$

is a basic closed set in $(2^U, Scott^{-1})$, since the disjunction is finite. Hence $\bigcap_{Z \in \mathcal{Z}} \{S : Z \cap S \neq \emptyset\}$ is closed. To show that

$$\widetilde{\mathcal{X}^{\supseteq}} \subseteq \bigcap_{Z \in \mathcal{Z}} \{S : Z \cap S \neq \emptyset\},$$

it will suffice then to show that

$$\mathcal{X}^{\supseteq} \subseteq \bigcap_{Z \in \mathcal{Z}} \{S : Z \cap S \neq \emptyset\}.$$

So let $S' \in \mathcal{X}^{\supseteq}$, which means that $S' \supseteq X'$ for some $X' \in \mathcal{X}$. Let $Z' \in \mathcal{Z}$. By definition of $\mathcal{Z}$, $Z' \cap X' \neq \emptyset$. Hence $Z' \cap S' \neq \emptyset$, so $S' \in \{S : Z' \cap S \neq \emptyset\}$, as required.

($\supseteq$) Now let $S' \in \bigcap_{Z \in \mathcal{Z}} \{S : Z \cap S \neq \emptyset\}$. Let $\mathcal{W}$ be a basic open set in $(2^U, Scott^{-1})$ such that $S' \in \mathcal{W}$. We must show that there exists some element $Y \in \mathcal{X}^{\supseteq}$ such that $Y \in \mathcal{W}$.

Suppose then that W is the finite subset of U which defines $\mathcal{W}$; that is,

$$\mathcal{W} = \{S : W \cap S = \emptyset\}.$$

Since $S' \in \mathcal{W}$, it is the case that $W \cap S' = \emptyset$. For every $Z \in \mathcal{Z}$, however, $Z \cap S' \neq \emptyset$, so we see that $W \notin \mathcal{Z}$. Hence, by definition of $\mathcal{Z}$, there exists $X' \in \mathcal{X}$ such that $W \cap X' = \emptyset$. Then this set X' satisfies the requirements for Y, since $X' \in \mathcal{X}^{\supseteq}$ and $X' \in \mathcal{W}$. ■

Algorithm 2.2 (for obtaining from minimal class $\mathcal{M}$ a prerequisite–free program P' such that Min(P') = $\mathcal{M}$)

Let $\mathcal{M} \subseteq 2^U$ be a minimal class. By Theorem 2.1, $\mathcal{M}$ is an antichain and $\mathcal{M}^{\supseteq}$ is closed in $(2^U, Scott^{-1})$.

If $\mathcal{M}$ is finite, then $\mathcal{M}^{\supseteq} = \bigcup_{X \in \mathcal{M}}\{S : X \subseteq S\}$ is a finite union of sets each defined by a countable conjunction. Use the distributive property to express $\mathcal{M}^{\supseteq}$ as a countable intersection of sets each defined by a finite disjunction. Since each disjunct is positive, this provides an expression for $\mathcal{M}^{\supseteq}$ as an intersection of sets which are basic closed in $(2^U, Scott^{-1})$.

In general, however, the distributive property does not suffice to obtain such an expression, since the disjunctions arrived at in this way are in general not finite. In this case, we may express $\mathcal{M}^{\supseteq} = \widetilde{\mathcal{M}^{\supseteq}}$ according to the preceding lemma as

$$\mathcal{M}^{\supseteq} = \bigcap_{Z \in \mathcal{Z}} \{S : Z \cap S \neq \emptyset\},$$

where

$$\mathcal{Z} = \{Z \subseteq U : Z \text{ is finite and for all X} \in \mathcal{M}, \mathrm{Z} \cap \mathrm{X} \neq \emptyset\}.$$

Again we arrive at an expression for $\mathcal{M}^{\supseteq}$ as an intersection of basic closed sets.

For each basic closed set in this expression, form a prerequisite–free clause from the disjunction defining the set, selecting one of its disjuncts (say, the first) to be the head. Let P' be the program consisting of all these clauses. As in Theorem 1.1, $Mod(P') = \mathcal{M}^{\supseteq}$, so $Min(P') = \min \mathcal{M}^{\supseteq}$. Then by Proposition 2.1, $\min \mathcal{M}^{\supseteq} = \min \mathcal{M} = \mathcal{M}$, since $\mathcal{M}$ is an antichain. ■

We now present a range of examples to illustrate the ideas of this section:

Example 2.3. Let P be the program

$$q \leftarrow \neg p$$
$$r \leftarrow q$$

which is not prerequisite–free. The collection $\mathcal{F}$ of models of P is $\mathcal{F}_1 \cap \mathcal{F}_2$, where

$$\mathcal{F}_1 = \{S : p \in S \vee q \in S\}$$
$$\mathcal{F}_2 = \{S : q \notin S \vee r \in S\}.$$

The key element of our method is to find an expression for $\mathcal{F}^{\supseteq}$. We proceed as in Algorithm 2.1:

(1) By the distributive property,

$$\begin{aligned}\mathcal{F} &= \mathcal{F}_1 \cap \mathcal{F}_2 \\ &= \{S : (p \in S \vee q \in S) \wedge (q \notin S \vee r \in S)\} \\ &= \{S : p \in S \wedge q \notin S\} \cup \{S : q \in S \wedge q \notin S\} \\ &\quad \cup \{S : p \in S \wedge r \in S\} \cup \{S : q \in S' \wedge r \in S\}.\end{aligned}$$

(2) Since the second summand is empty, we omit it:

$$\mathcal{F} = \{S : p \in S \wedge q \notin S\} \cup \{S : p \in S \wedge r \in S\} \cup \{S : q \in S \wedge r \in S\}.$$

(3) We form $\mathcal{F}^{\supseteq}$ by applying $\supseteq$ to each summand:

$$\mathcal{F}^{\supseteq} = \{S : p \in S \wedge q \notin S\}^{\supseteq} \cup \{S : p \in S \wedge r \in S\}^{\supseteq} \cup \{S : q \in S \wedge r \in S\}^{\supseteq}.$$

(4) We evaluate the application of $\supseteq$ to each summand by omitting the negative conjuncts:

$$\mathcal{F}^{\supseteq} = \{S : p \in S\} \cup \{S : p \in S \wedge r \in S\} \cup \{S : q \in S \wedge r \in S.$$

(5) Since the second summand is a subset of the first, it can be omitted, leaving only maximal summands:

$$\mathcal{F}^{\supseteq} = \{S : p \in S\} \cup \{S : q \in S \wedge r \in S\}.$$

According to Algorithm 2.1, we can read off the minimal models of P directly from this expression as

$$\mathcal{M} = \{\{p\}, \{q, r\}\}.$$

As in Algorithm 2.2, we can proceed further to find an expression for $\mathcal{F}^{\supseteq}$ as an intersection of basic closed sets:

(6) Since the union in Step (5) is finite, we can apply the distributive property to get the desired form:

$$\mathcal{F}^{\supseteq} = \{S : p \in S \vee q \in S\} \cap \{S : p \in S \vee r \in S\}.$$

Then the clauses

$$p \leftarrow \neg q$$

$$p \leftarrow \neg r$$

constitute a prerequisite–free program P' for which $Mod(P') = \mathcal{F}^{\supseteq}$ and hence $Min(P') = \mathcal{M} = Min(P)$.

Finally, from the expression in Step (6) we can construct, as in Theorem 2.3, the sentence $\mu =$

$$\begin{aligned} & (p \in S \leftrightarrow q \notin S \vee r \notin S) \\ \wedge \quad & (q \in S \leftrightarrow p \notin S) \\ \wedge \quad & (r \in S \leftrightarrow p \notin S) \end{aligned}$$

defining the minimal models of P. ■

Example 2.4 (More than one entry in body of clause). Let P be the program

$$u_2 \leftarrow \neg u_1$$
$$u_5 \leftarrow \neg u_2, \neg u_3, \neg u_4.$$

Since P is itself prerequisite–free, the set $Mod(P) =$

$$\{S : u_1 \in S \vee u_2 \in S\} \cap \{S : u_2 \in S \vee u_3 \in S \vee u_4 \in S \vee u_5 \in S\}$$

is closed in $(2^U, Scott^{-1})$. By Theorem 2.3, its minimal elements are defined by the sentence $\mu =$

$$\begin{array}{ll} & [u_1 \in S \leftrightarrow u_2 \notin S] \\ \wedge & [u_2 \in S \leftrightarrow u_1 \notin S \vee (u_3 \notin S \wedge u_4 \notin S \wedge u_5 \notin S)] \\ \wedge & [u_3 \in S \leftrightarrow (u_2 \notin S \wedge u_4 \notin S \wedge u_5 \notin S)] \\ \wedge & [u_4 \in S \leftrightarrow (u_2 \notin S \wedge u_3 \notin S \wedge u_5 \notin S)] \\ \wedge & [u_5 \in S \leftrightarrow (u_2 \notin S \wedge u_3 \notin S \wedge u_4 \notin S)]. \end{array}$$

To find its minimal models explicitly, we use Algorithm 2.1:

(1) By the distributive property, $Mod(P) =$

$$\begin{array}{llll} & \{S : u_1 \in S \wedge u_2 \in S\} & \cup & \{S : u_1 \in S \wedge u_3 \in S\} \\ \cup & \{S : u_1 \in S \wedge u_4 \in S\} & \cup & \{S : u_1 \in S \wedge u_5 \in S\} \\ \cup & \{S : u_2 \in S \wedge u_2 \in S\} & \cup & \{S : u_2 \in S \wedge u_3 \in S\} \\ \cup & \{S : u_2 \in S \wedge u_4 \in S\} & \cup & \{S : u_2 \in S \wedge u_5 \in S\}. \end{array}$$

Steps (2), (3), and (4) for evaluating $(Mod(P))^2$ leave this expression unchanged (since P is prerequisite–free).

(5) Omitting unnecessary terms, we get $(Mod(P))^2 =$

$$\begin{array}{l} \{S : u_2 \in S\} \cup \{S : u_1 \in S \wedge u_3 \in S\} \cup \\ \{S : u_1 \in S \wedge u_4 \in S\} \cup \{S : u_1 \in S \wedge u_5 \in S\}. \end{array}$$

Thus $Min(P) = \{\{u_2\}, \{u_1, u_3\}, \{u_1, u_4\}, \{u_1, u_5\}\}$. ■

Example 2.5 (Special case where $Min(P) = \{\emptyset\}$). Let P consist of the single clause

$$q \leftarrow p.$$

Since the set

$$Mod(P) = \{S : p \notin S \vee q \in S\}$$

is not closed in $(2^U, Scott^{-1})$, we pass to

$$(Mod(P))^{\supseteq} = \{S : p \notin S\}^{\supseteq} \cup \{S : q \in S\}^{\supseteq} = 2^U.$$

Now 2^U is the intersection of an empty collection of basic closed sets (or, in logic programming terms, is the set of models of the empty program P'). So according to Theorem 2.3, the minimal elements are defined by the sentence

$$\begin{aligned}\mu &= \bigwedge_{u \in U} (u \in S \leftrightarrow \bot)\\ &= \bigwedge_{u \in U} u \notin S.\end{aligned}$$

Thus the only minimal model of P is the empty subset of U. ■

Example 2.6 (Special case where *Min*(*P*) = {*U*}). Let P be the program consisting of all clauses of the form

$$u \leftarrow$$

for $u \in U$. The set of its models is naturally expressed as a set closed in $(2^U, Scott^{-1})$ by

$$Mod(P) = \bigcap_{u \in U} \{S : u \in S\}.$$

According to Theorem 2.3, its minimal elements are defined by the sentence

$$\begin{aligned}\mu &= \bigwedge_{u \in U} (u \in S \leftrightarrow \top)\\ &= \bigwedge_{u \in U} u \in S.\end{aligned}$$

Thus U itself, the only subset of U which satisfies μ, is the only minimal model of P. ■

Example 2.7. Let P be the infinite program

$$\begin{gathered}\mathrm{u}_1 \leftarrow \neg u_0\\ u_3 \leftarrow \neg u_2\\ \vdots\\ u_{2i+1} \leftarrow \neg u_{2i}\\ \vdots\\ u_2 \leftarrow u_0\end{gathered}$$

where $U = \{u_0, u_1, u_2, \ldots\}$. Since P is not prerequisite–free, we pass to $(Mod(P))^{\supseteq}$, which we can compute according to Algorithm 2.1. To simplify the notation, let

$$\Psi = \{\psi \in \omega^\omega : \psi(i) = 2i \text{ or } \psi(i) = 2i + 1\}.$$

Then $[Mod(P)]^{\supseteq}$

$$\begin{aligned}
&= [\{S : u_0 \notin S \vee u_2 \in S\} \cap \bigcap_{i\in\omega} \{S : u_{2i} \in S \vee u_{2i+1} \in S\}]^{\supseteq} \\
&= [\{S : u_0 \notin S \vee u_2 \in S\} \cap \bigcup_{\psi\in\Psi} \{S : \bigwedge_{i\in\omega} \psi(i) \in S\}]^{\supseteq} \\
&= \bigcup_{\psi\in\Psi} \{S : u_0 \notin S \wedge \bigwedge_{i\in\omega} \psi(i) \in S\}^{\supseteq} \cup \bigcup_{\psi\in\Psi} \{S : u_2 \in S \wedge \bigwedge_{i\in\omega} \psi(i) \in S\}^{\supseteq} \\
&= \bigcup_{\psi\in\Psi} \{S : u_1 \in S \wedge u_2 \in S \wedge \bigwedge_{i\geq 2} \psi(i) \in S\} \cup \\
&\quad \bigcup_{\psi\in\Psi} \{S : u_1 \in S \wedge u_3 \in S \wedge \bigwedge_{i\geq 2} \psi(i) \in S\} \cup \\
&\quad \bigcup_{\psi\in\Psi} \{S : u_0 \in S \wedge u_2 \in S \wedge \bigwedge_{i\geq 2} \psi(i) \in S\}.
\end{aligned}$$

Then we can read off from this expression that the minimal elements of $Mod(P)$ are those sets which consist of u_0 and u_2 and one element from each pair $\{u_4, u_5\}$, $\{u_6, u_7\}, \ldots$, together with those sets which consist of u_1 and either u_2 or u_3 and one element from each pair $\{u_4, u_5\}$, $\{u_6, u_7\}, \ldots$.

As a matter of interest, we may proceed further using Algorithm 2.2 to find a prerequisite–free program P' with the same minimal models. In this case, however, the distributive property applied to the preceding expression for $(Mod(P))^{\supseteq}$ doesn't produce the proper form, nor is it evident how to factor this expression as a product of finite factors. Instead, as in Algorithm 2.2, we look for those finite sets $Z \subseteq U$ such that every minimal model of P contains some element of Z. There are no singleton sets with this property. Among the sets with two elements, every pair $\{u_{2i}, u_{2i+1}\}$ is of this sort; that is, for all $X \in Min(P)$, $u_{2i} \in X$ or $u_{2i+1} \in X$. Beyond these, only the pair $\{u_1, u_2\}$ has this property. Conditions involving sets Z of three or more elements turn out in this instance to be redundant. As in Algorithm 2.2, then,

$$\mathcal{M}^{\supseteq} = \{S : u_1 \in S \vee u_2 \in S\} \cap \bigcap_{i\geq 0} \{S : u_{2i} \in S \vee u_{2i+1} \in S\},$$

where $\mathcal{M}$ is the set of minimal models of P described above. Translating the basic closed sets into clauses, we obtain finally the following prerequisite–free program P'

with $Min(P') = \mathcal{M}$:

$$\begin{aligned} u_2 &\leftarrow \neg u_1 \\ u_1 &\leftarrow \neg u_0 \\ u_3 &\leftarrow \neg u_2 \\ &\vdots \\ u_{2i+1} &\leftarrow \neg u_{2i} \\ &\vdots \end{aligned}$$

■

Example 2.8. Predicate logic programs can be handled similarly, upon replacement of each clause by its set of ground instances. Consider the following predicate logic program P, based on the preceding example:

$$\begin{aligned} &even(0) \\ even(\sigma^2(X)) &\leftarrow even(X) \\ u(\sigma(X)) &\leftarrow even(X), \neg u(X) \\ u(\sigma^2(0)) &\leftarrow u(0), \end{aligned}$$

with $ground(P)$ then consisting of

$$\begin{aligned} &even(0) \\ even(\sigma^2(0)) &\leftarrow even(0) \\ even(\sigma^3(0)) &\leftarrow even(\sigma(0)) \\ even(\sigma^4(0)) &\leftarrow even(\sigma^2(0)) \\ &\vdots \\ u(\sigma(0)) &\leftarrow even(0), \neg u(0) \\ u(\sigma^2(X)) &\leftarrow even(\sigma(0)), \neg u(\sigma(0)) \\ u(\sigma^3(X)) &\leftarrow even(\sigma^2(0)), \neg u(\sigma^2(0)) \\ &\vdots \\ u(\sigma^2(0)) &\leftarrow u(0). \end{aligned}$$

By the method above, we find that the minimal models are those subsets of the

Herbrand base which consist of

$$even(0), even(\sigma^2(0)), even(\sigma^4(0)), \ldots$$

and either $u(0)$ and $u(\sigma^2(0))$ and one element from each pair

$$\{u(\sigma^4(0)), u(\sigma^5(0))\}, \{u(\sigma^6(0)), u(\sigma^7(0))\}, \ldots,$$

or $u(\sigma(0))$ and $u(\sigma^2(0))$ or $u(\sigma^3(0))$ and one element from each pair

$$\{u(\sigma^4(0)), u(\sigma^5(0))\}, \{u(\sigma^6(0)), u(\sigma^7(0))\}, \ldots.$$

The calculation is similar to that of the last example, though somewhat messier. ■

3. Stable model classes

One cannot help but notice the formal resemblance between the sentence μ developed here which characterizes the minimal models of a program P and the sentence σ developed, as mentioned before, by Marek, Nerode, and Remmel, which characterizes its stable models. Recall that μ has the form

$$\bigwedge_{u\in U}[u \in S \leftrightarrow \bigvee_{A\in\Phi_u} \vec{A} \notin S]$$

and σ has the form

$$\bigwedge_{u\in U}[u \in S \leftrightarrow \bigvee_{A\in\Psi_u} \vec{A} \notin S]$$

for certain collections Φ_u and Ψ_u of finite sets $A \subseteq U$. As concerns μ, the negative information pertaining to element $u \in U$ comes from the supports of u in the clauses of a prerequisite–free program P' associated with P. To give a similar description concerning σ, we first make explicit a "normal–form" result for stable model classes which is implicit in [4] and can also be derived from [2]:

Proposition 3.1 (Prerequisite–free representation) [4]. *For any propositional logic program P, there is a prerequisite–free program $P^\dagger$ such that $Stab(P) = Stab(P^\dagger)$.*

Proof. Let

$$\bigwedge_{u\in U}[R(\underline{u}) \leftrightarrow \bigvee_{A\in\Psi_u} \bigwedge_{a\in A} \neg R(\underline{a})]$$

be the sentence σ which specifies the stable models of P, so that $(U, S, \vec{u}_{u \in U}) \models \sigma$ if and only if

$$\bigwedge_{u \in U} (u \in S \leftrightarrow \bigvee_{A \in \Psi_u} \vec{A} \notin S).$$

Recall that Ψ_u is the set of supports of minimal proof schemes for u over program P. Let $P^\dagger$ be the program consisting of all clauses

$$u \leftarrow \neg\vec{A}$$

for $u \in U$ and $A \in \Psi_u$. For each $u \in U$, the supports of proof schemes for u in $P^\dagger$ are exactly the finite sets $A \in \Psi_u$. So the sentence $\sigma^\dagger$ which specifies the stable models of $P^\dagger$ is σ itself. Thus

$$Stab(P) = \{S : (U, S, \vec{u}_{u \in U}) \models \sigma\} = Stab(P^\dagger). \qquad \blacksquare$$

For the sentence σ, then, the negative information pertaining to element $u \in U$ also amounts to the supports of u in certain prerequisite–free clauses, namely those clauses of the associated program $P^\dagger$ which have u as their head. Suppose that P is itself a prerequisite–free program; then $P = P' = P^\dagger$. For each u, the collection Ψ_u (of supports of u in clauses where u is the head) is a subset of Φ_u (the supports of u in clauses in which u appears as the head or in the body). So in this case the sentence σ is actually a subexpression of the sentence μ.

We can use this comparison of defining equations for minimal and stable model classes to make a comparison of the classes themselves. Based on the characterization of minimal model classes in Theorem 2.1, we will then be able to characterize stable model classes as well. Because we will be comparing classes associated with different programs, we use subscript P to identify the defining equations associated with program P.

To begin with, consider an arbitrary minimal model class, say the minimal models of program P. Let P' be a prerequisite–free program for which $\mu_P = \mu_{P'}$. Form a new program $P^\sharp$ from P' as follows: For every clause ρ in P', include in $P^\sharp$ all the clauses formed by rearranging ρ so that different atoms from ρ appear as the head. Then $\mu_{P'} = \sigma_{P^\sharp}$. So the class of minimal models of P can be realized as the class of stable models of a related program $P^\sharp$. By a considerable generalization of this approach, we can prove the following sufficient condition for a class to be realizable as the stable models of some program:

Theorem 3.1. *Let P be a general logic program over U. Let $\mathcal{S}$ be a nonempty G_δ subset (in the inverse–Scott sense) of $Min(P)$. Then there is a general logic program Γ over U such that $\mathcal{S} = Stab(\Gamma)$.*

Proof. Let $U = \{u_k : k < |U|\}$. We have that $\mathcal{S} = Min(P) \cap \bigcap_{k\in\omega} \mathcal{G}_k$ for open subsets $\mathcal{G}_k = \{S : \bigvee_{B\in\mathcal{B}_k} \vec{B} \not\subseteq S\}$ of $(2^U, Scott^{-1})$, where each $\mathcal{B}_k$, recall, is a collection of finite subsets of U. By Theorem 2.3, for each $k < |U|$ there is a set Φ_{u_k} of finite subsets of U such that $Min(P) = \{S : \bigwedge_{k<|U|}[u_k \in S \leftrightarrow \bigvee_{A\in\Phi_{u_k}} \vec{A} \not\subseteq S]\}$.

For each $k \in \omega$, let Γ_k consist of all clauses of the form

$$u_k \leftarrow \neg\vec{A}, \neg\vec{B}^1, \ldots, \neg\vec{B}^k$$

for $A \in \Phi_{u_k}, B^1 \in \mathcal{B}_1, \ldots, B^k \in \mathcal{B}_k$, if $k < |U|$ and Φ_{u_k} is nonempty; otherwise, let $\Gamma_k = \emptyset$. (Note that every clause in Γ_k has u_k as its head, and that the entries in the body are drawn only from the definitions of $\mathcal{G}_1, \ldots, \mathcal{G}_k$ and the kth conjunct of μ_P.)

It may happen that the set $\mathcal{B}_k$ of sequences which defines $\mathcal{G}_k$ has a finite hit set—that is, a set $\{c_1, \ldots, c_n\}$ such that $\{c_1, \ldots, c_n\} \cap B$ is nonempty for each $B \in \mathcal{B}_k$. To make it impossible for a finite set containing such a hit set to be a stable model, we form the set Ξ_k consisting of all clauses

$$z \leftarrow c_1, \ldots, c_n,$$

where $\{c_1, \ldots, c_n\}$ is a minimal finite hit set for $\mathcal{B}_k$ and $z \in U$.

Now let $\Gamma = \bigcup_{k\in\omega}(\Gamma_k \cup \Xi_k)$. We will show that $\mathcal{S} = Stab(\Gamma)$.

($\subseteq$) Let $S \in \mathcal{S}$. We will show that $S \in Stab(\Gamma)$ by showing that $S = T_{GL_S(\Gamma)} \uparrow \omega(\emptyset)$.

(a) First we show that $S \subseteq T_{GL_S(\Gamma)} \uparrow \omega(\emptyset)$:

If $S = \emptyset$, the statement is true.

So suppose that $S \neq \emptyset$, and let $u_\ell \in S$. Since $S \in Min(P)$, according to μ_P we have $\vec{A} \not\subseteq S$ for some $A \in \Phi_{u_\ell}$. Since $S \in \bigcap_{k\in\omega} \mathcal{G}_k$, we have also $\vec{B}^1 \not\subseteq S, \vec{B}^2 \not\subseteq S, \ldots$ for some $B^1 \in \mathcal{B}_1, B^2 \in \mathcal{B}_2, \ldots$.

In particular, no element of the sequence $\vec{A}\vec{B}^1 \ldots \vec{B}^\ell$ belongs to S.

Then since the clause

$$u_\ell \leftarrow \neg\vec{A}, \neg\vec{B}^1, \ldots, \neg\vec{B}^\ell$$

belongs to Γ, its S-reduct

$$u_\ell \leftarrow$$

belongs to $GL_S(\Gamma)$. So $u_\ell \in T_{GL_S(\Gamma)} \uparrow 1(\emptyset)$.

(b) Now we show by induction on n that $T_{GL_S(\Gamma)} \uparrow n(\emptyset) \subseteq S$:

Since $T_{GL_S(\Gamma)} \uparrow 0(\emptyset) = \emptyset$, the statement is true for $n = 0$.

Now suppose that $u_\ell \in T_{GL_S(\Gamma)} \uparrow n + 1(\emptyset)$. Then u_ℓ is the head of some Horn clause ρ in $GL_S(\Gamma)$ for which every entry in the body belongs to $T_{GL_S(\Gamma)} \uparrow n(\emptyset)$.

We claim that the clause

$$u_\ell \leftarrow$$

belongs to $GL_S(\Gamma)$. For otherwise there is a clause

$$u_\ell \leftarrow c_1, \ldots, c_n$$

in $\bigcup_{k\in\omega} \Xi_k$ for which $\{c_1, \ldots, c_n\} \subseteq T_{GL_S(\Gamma)} \uparrow n(\emptyset)$, and hence, by the induction hypothesis, $\{c_1, \ldots, c_n\} \subseteq S$. Since $\{c_1, \ldots, c_n\}$ is a hit set for some $\mathcal{B}_k$, S meets every $B \in \mathcal{B}_k$, which contradicts the assumption that $S \in \mathcal{G}_k$.

Because the clause

$$u_\ell \leftarrow$$

is in $GL_S(\Gamma)$, there is a clause

$$u_\ell \leftarrow \neg\vec{A}, \neg\vec{B^1}, \ldots, \neg\vec{B^\ell}$$

in Γ_ℓ with $A \in \Phi_{u_\ell}$ such that $\vec{A} \notin S$. Since by assumption $S \in Min(P)$ and hence satisfies μ_P, we conclude that $u_\ell \in S$, as desired.

($\supseteq$) Now let $S \in Stab(\Gamma)$. We will show that $S \in \mathcal{S}$ by showing that (a) $S \in \bigcap_{k\in\omega} \mathcal{G}_k$ and (b) $S \in Min(P)$.

We observe first that for each k there is no finite hit set $\{c_1, \ldots, c_n\}$ for $\mathcal{B}_k$ for which $\{c_1, \ldots, c_n\} \subseteq S$. For otherwise every element of U would belong to $T_{GL_S(\Gamma)} \uparrow \omega(\emptyset) = S$ by virtue of the clauses

$$z \leftarrow c_1, \ldots, c_n$$

in $GL_S(\Gamma)$. But $T_{GL_U(\Gamma)} \uparrow \omega(\emptyset) = \emptyset \neq U$, so U is not a stable model of Γ, which contradicts $U = S$.

Because S is a stable model of Γ, it then follows that, for every element u_k of S, the clause

$$u_k \leftarrow$$

appears in $GL_S(\Gamma)$.

(a) Let ℓ be given. We show that $S \in \mathcal{G}_\ell$:

Case 1. If S is infinite, then there exists $m > \ell$ such that $u_m \in S$. Since the clause

$$u_m \leftarrow$$

appears in $GL_S(\Gamma)$, there is then a clause

$$u_m \leftarrow \neg\vec{A}, \neg\vec{B^1}, \ldots, \neg\vec{B^m}$$

in Γ_m for which $\vec{A} \notin S, \vec{B^1} \notin S, \ldots, \vec{B^m} \notin S$. In particular, $\vec{B^\ell} \notin S$ for some $B^\ell \in \mathcal{B}_\ell$. Therefore $S \in \mathcal{G}_\ell$.

Case 2. If S is finite, then, because no finite subset is a hit set for $\mathcal{B}_\ell$, the same property holds for S itself. That is, $S \cap B^\ell = \emptyset$ for some $B^\ell \in \mathcal{B}_\ell$. So $S \in G_\ell$.
(b) Let ℓ be given. We show that S satisfies the ℓth conjunct

$$u_\ell \in S \leftrightarrow \bigvee_{A \in \Phi_{u_\ell}} \vec{A} \notin S$$

of μ_P :
→ Suppose that $u_\ell \in S$. There is then some clause

$$u_\ell \leftarrow \neg\vec{A}, \neg\vec{B}^1, \ldots, \neg\vec{B}^\ell$$

in Γ_ℓ with $A \in \Phi_{u_\ell}$ corresponding to the clause

$$u_\ell \leftarrow$$

in $GL_S(\Gamma)$. In particular, $\vec{A} \notin S$ for this value of $A \in \Phi_{u_\ell}$.
← Suppose that $\vec{A} \notin S$ for some $A \in \Phi_{u_\ell}$. From part (a) we know that $S \in \bigcap_{k \in \omega} \mathcal{G}_k$, so for each $k \in \omega$ there is some $B^k \in \mathcal{B}_k$ such that $B^k \notin S$. Then the clause

$$u_\ell \leftarrow$$

appears in $GL_S(\Gamma)$, corresponding to the clause

$$u_\ell \leftarrow \neg\vec{A}, \neg\vec{B}^1, \ldots, \neg\vec{B}^\ell$$

in Γ_ℓ. Therefore $u_\ell \in T_{GL_S(\Gamma)} \uparrow \omega(\emptyset) = S$, as desired. ■

Example 3.1. Consider again the program P

$$u_2 \leftarrow \neg u_1$$
$$u_5 \leftarrow \neg u_2, \neg u_3, \neg u_4$$

of Example 2.4 above. As determined earlier, its minimal models are $\{u_2\}$, $\{u_1, u_3\}$, $\{u_1, u_4\}$, and $\{u_1, u_5\}$. According to the theorem, any inverse–Scott G_δ subset of this collection $\mathcal{M}$ is expressible as the set of stable models of an associated program Γ. Let us check the prescription for Γ when the subset to be represented is $\mathcal{M}$ itself.

As determined before, the sentence μ_P which characterizes $Min(P)$ is

$$\begin{aligned}
& [u_1 \in S \leftrightarrow u_2 \notin S] \\
\wedge\ & [u_2 \in S \leftrightarrow u_1 \notin S \vee (u_3 \notin S \wedge u_4 \notin S \wedge u_5 \notin S)] \\
\wedge\ & [u_3 \in S \leftrightarrow (u_2 \notin S \wedge u_4 \notin S \wedge u_5 \notin S)] \\
\wedge\ & [u_4 \in S \leftrightarrow (u_2 \notin S \wedge u_3 \notin S \wedge u_5 \notin S)] \\
\wedge\ & [u_5 \in S \leftrightarrow (u_2 \notin S \wedge u_3 \notin S \wedge u_4 \notin S)].
\end{aligned}$$

For this case, we have $\mathcal{S} = Min(P) \cap \mathcal{G}$, where we take $\mathcal{G} = 2^U$. As in the theorem, we express $\mathcal{G}$ as $\{S : \bigvee_{B \in \emptyset} \vec{B} \not\subseteq S\}$.
Then we form the clauses of Γ_k for $k \in \omega$ as follows:

$$\Gamma_1 = \{u_1 \leftarrow \neg u_2\}$$

$$\Gamma_2 = \{u_2 \leftarrow \neg u_1, \quad u_2 \leftarrow \neg u_3, \neg u_4, \neg u_5\}$$

$$\Gamma_3 = \{u_3 \leftarrow \neg u_2, \neg u_4, \neg u_5\}$$

$$\Gamma_4 = \{u_4 \leftarrow \neg u_2, \neg u_3, \neg u_5\}$$

$$\Gamma_5 = \{u_5 \leftarrow \neg u_2, \neg u_3, \neg u_4\}$$

$$\Gamma_k = \emptyset \text{ for } k \geq 6.$$

There are no finite hit sets for $\{B : B \in \{\emptyset\}\}$, so we form no additional clauses in the sets Ξ_k.
Let Γ then be $\bigcup_{k=1}^{5} \Gamma_k$.
By inspection, $Stab(\Gamma) = \{\{u_2\}, \{u_1, u_3\}, \{u_1, u_4\}, \{u_1, u_5\}\}$, as desired. ■

We consider next whether the condition shown in Theorem 3.1 to be sufficient for $\mathcal{S}$ to be a stable model class—that $\mathcal{S}$ be a G_δ subset (in the inverse–Scott sense) of a minimal model class—is also necessary. Certainly the set of stable models of a given program P is a subset of the minimal model class of P itself, and moreover is known to be a G_δ set—but in the wrong topology. For if

$$\sigma_P = \bigwedge_{u \in U} (u \in S \leftrightarrow \bigvee_{A \in \Psi_u} \bigwedge_{a \in A} a \notin S)$$

is the usual sentence specifying the stable models of P, then $Stab(P)$ is an intersection over $u \in U$ of sets of the form

$$\begin{aligned} & \{S : u \in S \leftrightarrow \textstyle\bigvee_{A \in \Psi_u} \bigwedge_{a \in A} a \notin S\} \\ = \;& \{S : u \in S \rightarrow \textstyle\bigvee_{A \in \Psi_u} \bigwedge_{a \in A} a \notin S\} \cap \{S : (\bigvee_{A \in \Psi_u} \bigwedge_{a \in A} a \notin S) \rightarrow u \in S\} \\ = \;& \{S : u \notin S \vee \textstyle\bigvee_{A \in \Psi_u} \bigwedge_{a \in A} a \notin S\} \cap \{S : (\bigwedge_{A \in \Psi_u} \bigvee_{a \in A} a \in S) \vee u \in S\}. \end{aligned}$$

In both topologies, the first of these factors is open and the second closed. The second is also a G_δ set in $(2^U, Cantor)$, but not in general in $(2^U, Scott^{-1})$. In fact, the only closed sets which are also G_δ sets in the inverse–Scott topology are $\emptyset$ and 2^U.

Fortunately, the condition of Theorem 3.1 does not require that $\mathcal{S}$ itself be a G_δ subset of $(2^U, Scott^{-1})$, but only that it be the intersection of a G_δ set with a minimal model class. In the following theorem, we use our earlier observation about the relation between σ and μ for prerequisite–free programs to show that every stable model class does indeed have this property. We state this as a characterization theorem, along with some further equivalences along the line of Theorem 2.1:

Theorem 3.2 (Characterization of stable model classes (I)). *For $\mathcal{S} \subseteq 2^U$ with the inverse–Scott topology, the following are equivalent:*

(i) *$\mathcal{S}$ is a stable model class.*

(ii) *$\mathcal{S}$ is a G_δ subset of a minimal model class.*

(iii) *$\mathcal{S}$ is a G_δ subset of* $\min \tilde{\mathcal{S}}$.

(iv) *$\mathcal{S}$ is a G_δ subset of* $\min \bar{\mathcal{S}}$.

Proof. If $\mathcal{S} = \emptyset$, then the theorem certainly holds. So we assume throughout that $\mathcal{S} \neq \emptyset$.

(i) $\Rightarrow$ (ii)
Suppose that $\mathcal{S} = \mathit{Stab}(P)$. Pass to the prerequisite–free program $P^\dagger$ which has the same stable models as P. Recall that

$$\sigma_{P^\dagger} = \bigwedge_{u \in U} [u \in S \leftrightarrow \bigvee_{A \in \Psi_u} \vec{A} \notin S]$$

is a subexpression of

$$\mu_{P^\dagger} = \bigwedge_{u \in U} [u \in S \leftrightarrow \bigvee_{A \in \Phi_u} \vec{A} \notin S]$$

in the sense that $\Psi_u \subseteq \Phi_u$ for each $u \in U$. Then

$$\begin{aligned}
&\mathit{Stab}(P)\\
&\quad = \{S : (U, S, \vec{u}_{u \in U}) \models \sigma_{P^\dagger}\}\\
&\quad = \{S : \bigwedge_{u \in U} [u \in S \leftrightarrow \bigvee_{A \in \Psi_u} \vec{A} \notin S]\}\\
&\quad = \{S : \bigwedge_{u \in U} [(u \in S \leftrightarrow \bigvee_{A \in \Phi_u} \vec{A} \notin S) \wedge (u \in S \rightarrow \bigvee_{A \in \Psi_u} \vec{A} \notin S)]\}\\
&\quad = \{S : (U, S, \vec{u}_{u \in U}) \models \mu_{P^\dagger}\} \cap \{S : \bigwedge_{u \in U} [u \in S \rightarrow \bigvee_{A \in \Psi_u} \vec{A} \notin S]\}\\
&\quad = \mathit{Min}(P^\dagger) \cap \bigcap_{u \in U} \{S : x \notin S \vee \bigvee_{A \in \Psi_u} \vec{A} \notin S\}.
\end{aligned}$$

We note that each set $\{S : u \notin S \vee \bigvee_{A \in \Psi_u} \vec{A} \notin S\}$ is open.

(ii) ⇒ (iii)
Given that $\mathcal{S}$ is a G_δ subset of $Min(P)$ for some P, let $\mathcal{F}_0 = Mod(P)$ and $\mathcal{F} = \mathcal{F}_0{}^{\supseteq}$. Then $\mathcal{S} = (\min \mathcal{F}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$, where $\mathcal{F}$ is closed and each $\mathcal{G}_k$ is open. We will show that $\mathcal{S} = (\min \tilde{\mathcal{S}}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$. We note first that $\mathcal{S} \subseteq \tilde{\mathcal{S}} \subseteq \mathcal{F}$.
($\subseteq$) Let $S \in \mathcal{S} = (\min \mathcal{F}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$.
By set theory we get that $S \in \min \tilde{\mathcal{S}}$, since $S' \in \tilde{\mathcal{S}}$ and $S' \subseteq S \Rightarrow S' \in \mathcal{F}$ and $S' \subseteq S \Rightarrow S' = S$.
Also by assumption $S \in \bigcap_{k\in\omega} \mathcal{G}_k$.
So $S \in (\min \tilde{\mathcal{S}}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$.
($\supseteq$) Let $S \in (\min \tilde{\mathcal{S}}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$.
Since $S \in \tilde{\mathcal{S}} \subseteq \mathcal{F}$, by Lemma 2.1 there exists $T \subseteq S$ such that $T \in \min \mathcal{F}$. Since open sets are specified by only negative information, $S \in \bigcap_{k\in\omega} \mathcal{G}_k \Rightarrow T \in \bigcap_{k\in\omega} \mathcal{G}_k$. Hence $T \in \mathcal{S}$. But because $S \in \min \tilde{\mathcal{S}}$, we have $T = S$, so $S \in \mathcal{S}$.

(iii) ⇒ (iv)
We are given that $\mathcal{S} = (\min \tilde{\mathcal{S}}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$ for open sets $\mathcal{G}_k$. We can show that $\mathcal{S} = (\min \bar{\mathcal{S}}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$ by the same argument as above (for (ii) ⇒ (iii)), but based on the relation $\mathcal{S} \subseteq \bar{\mathcal{S}} \subseteq \tilde{\mathcal{S}}$ in place of $\mathcal{S} \subseteq \tilde{\mathcal{S}} \subseteq \mathcal{F}$.

(iv) ⇒ (i)
We are given that $\mathcal{S}$ is a G_δ subset of $\min \bar{\mathcal{S}}$. Since $\mathcal{S} \neq \emptyset$, $\min \bar{\mathcal{S}} = \min(\bar{\mathcal{S}} \cup \{U\})$. Since $\bar{\mathcal{S}} \cup \{U\}$ is closed in the Cantor topology and has U as an element, by Theorem 1.1 there is a program P such that $\bar{\mathcal{S}} \cup \{U\} = Mod(P)$, hence $\min(\bar{\mathcal{S}} \cup \{U\}) = Min(P)$. So $\mathcal{S}$ is a G_δ subset of $Min(P)$. By Theorem 3.1 there is then a program Γ for which $\mathcal{S} = Stab(\Gamma)$. ■

From this we can also derive the following characterization of stable model classes which does not mention minimal model classes:

Theorem 3.3 (Characterization of stable model classes (II)). *For $\mathcal{S} \subseteq 2^U$ with the inverse–Scott topology, the following are equivalent:*

(i) *$\mathcal{S}$ is a stable model class.*
(ii) *$\mathcal{S}$ is an antichain and a G_δ subset of a closed set.*
(iii) *$\mathcal{S}$ is an antichain and a G_δ subset of $\tilde{\mathcal{S}}$.*
(iv) *$\mathcal{S}$ is an antichain and a G_δ subset of $\bar{\mathcal{S}}$.*

Proof. As before, we may assume that $\mathcal{S} \neq \emptyset$.

(i) ⇒ (ii)
Recall that in [4] it was shown that a stable model class forms an antichain. Suppose

now that $\mathcal{S} = Stab(P)$. Then

$$\begin{aligned}
\mathcal{S} &= \{S : (U, S, \vec{u}_{u\in U}) \models \sigma_P\} \\
&= \{S : \bigwedge_{u\in U} [u \in S \leftrightarrow \bigvee_{A\in\Psi_u} \vec{A} \not\subseteq S]\} \\
&= \bigcap_{u\in U} \{S : (\bigvee_{A\in\Psi_u} \vec{A} \not\subseteq S) \to u \in S\} \cap \bigcap_{u\in U} \{S : u \in S \to \bigvee_{A\in\Psi_u} \vec{A} \not\subseteq S\} \\
&= \bigcap_{u\in U} \{S : (\bigwedge_{A\in\Psi_u} \bigvee_{a\in A} a \in S) \vee u \in S\} \cap \bigcap_{u\in U} \{S : u \notin S \vee \bigvee_{A\in\Psi_u} \vec{A} \not\subseteq S\}.
\end{aligned}$$

We note that each set $\{S : u \notin S \vee \bigvee_{A\in\Psi_u} \vec{A} \not\subseteq S\}$ is open, and that the set $\bigcap_{u\in U}\{S : (\bigwedge_{A\in\Psi_u} \bigvee_{a\in A} a \in S) \vee u \in S\}$ is closed (and is in fact $Mod(P^{\dagger})$).

(ii) ⇒ (iii)
We are given that $\mathcal{S} = \mathcal{F} \cap \bigcap_{k\in\omega} \mathcal{G}_k$ for open sets $\mathcal{G}_k$ and closed set $\mathcal{F}$. We wish to show that $\mathcal{S} = \tilde{\mathcal{S}} \cap \bigcap_{k\in\omega} \mathcal{G}_k$. By definition of closure, $\mathcal{S} \subseteq \tilde{\mathcal{S}} \subseteq \mathcal{F}$, and by assumption $\mathcal{S} \subseteq \bigcap_{k\in\omega} \mathcal{G}_k$. So

$$\mathcal{S} \subseteq \tilde{\mathcal{S}} \cap \bigcap_{k\in\omega} \mathcal{G}_k \subseteq \mathcal{F} \cap \bigcap_{k\in\omega} \mathcal{G}_k = \mathcal{S}.$$

(iii) ⇒ (iv)
We are given that $\mathcal{S} = \tilde{\mathcal{S}} \cap \bigcap_{k\in\omega} \mathcal{G}_k$ for open sets $\mathcal{G}_k$. We can show that $\mathcal{S} = \bar{\mathcal{S}} \cap \bigcap_{k\in\omega} \mathcal{G}_k$ by the same argument as for (ii) ⇒ (iii), but based on the relation $\mathcal{S} \subseteq \bar{\mathcal{S}} \subseteq \tilde{\mathcal{S}}$ in place of $\mathcal{S} \subseteq \tilde{\mathcal{S}} \subseteq \mathcal{F}$.

(iv) ⇒ (i)
We are given that $\mathcal{S}$ is an antichain and $\mathcal{S} = \bar{\mathcal{S}} \cap \bigcap_{k\in\omega} \mathcal{G}_k$ for open sets $\mathcal{G}_k$. To show that $\mathcal{S}$ is a stable model class, by Theorem 3.2 it will suffice to show that $\mathcal{S} = (\min \bar{\mathcal{S}}) \cap \bigcap_{k\in\omega} \mathcal{G}_k$.
($\supseteq$) On the one hand, $(\min \bar{\mathcal{S}}) \cap \bigcap_{k\in\omega} \mathcal{G}_k \subseteq \bar{\mathcal{S}} \cap \bigcap_{k\in\omega} \mathcal{G}_k = \mathcal{S}$.
($\subseteq$) To show the other inclusion, let $S \in \mathcal{S}$. By assumption, $S \in \bigcap_{k\in\omega} \mathcal{G}_k$, so it remains to show that $S \in \min \bar{\mathcal{S}}$. Let T then be an element of $\bar{\mathcal{S}}$ with $T \subseteq S$. Since open sets are specified by negative information, $S \in \bigcap_{k\in\omega} \mathcal{G}_k \Rightarrow T \in \bigcap_{k\in\omega} \mathcal{G}_k$. Hence $T \in \mathcal{S}$. But because $\mathcal{S}$ is an antichain, $T = S$. Therefore $S \in \min \bar{\mathcal{S}}$, as desired. ■

With these results, we can now show by example that the antichain and Cantor G_δ properties mentioned in the Introduction are not sufficient to characterize stable model classes:

Example 3.2. Consider again the set $\mathcal{S} = \{\{0\}, \{1\}, \{2\}, \ldots\}$ of Example 2.1, which, as noted there, is both a G_δ set in the Cantor sense and an antichain. We see that $\min\bar{\mathcal{S}} = \{\emptyset\}$, since $\emptyset \in \bar{\mathcal{S}}$. Previously we applied Theorem 2.1 to show that $\mathcal{S}$ is not a minimal model class, because $\mathcal{S} \neq \min\bar{\mathcal{S}}$. But it is also the case that $\mathcal{S} \not\subseteq \min\bar{\mathcal{S}}$ (in fact $\mathcal{S} \cap \min\bar{\mathcal{S}} = \emptyset$!) So by Theorem 3.2, $\mathcal{S}$ also cannot be a stable model class. ■

Finally we use the characterization theorems for minimal and stable model classes to give an example of a stable model class which cannot be realized as a minimal model class:

Example 3.3. Let

$$\begin{aligned} S_0 &= \{u_{2n} : n \in \omega, n \neq 0\} \cup \{u_1\} \\ S_1 &= \{u_{2n} : n \in \omega, n \neq 1\} \cup \{u_3\} \\ S_2 &= \{u_{2n} : n \in \omega, n \neq 2\} \cup \{u_5\} \\ &\vdots \\ S_k &= \{u_{2n} : n \in \omega, n \neq k\} \cup \{u_{2k+1}\} \\ &\vdots \end{aligned}$$

Let $\mathcal{S} = \{S_k : k \in \omega\}$.
Then $\bar{\mathcal{S}} = \mathcal{S} \cup \{\{u_{2n} : n \in \omega\}\} = \min\bar{\mathcal{S}}$.
By Theorem 2.1, $\mathcal{S}$ is not a minimal model class, because $\mathcal{S} \neq \min\bar{\mathcal{S}}$.
However, $\mathcal{S}$ is an open subset of $\min\bar{\mathcal{S}}$ (or $\bar{\mathcal{S}}$), namely

$$\mathcal{S} = (\min\bar{\mathcal{S}}) \cap \{S : u_0 \notin S \vee u_2 \notin S \vee u_4 \notin S \vee \ldots\}.$$

So, by Theorem 3.2 (or Theorem 3.3), $\mathcal{S}$ is in fact a stable model class. ■

4. Conclusion

Our main goal in this investigation was to provide characterization theorems for stable and minimal model classes. Along the way we developed defining equations and prerequisite–free representation for minimal model classes, similar to results already known for stable model classes, which appear to be useful tools in their own right. Furthermore, we have developed a point of view which clarifies the distinction between stable and minimal models, in particular concerning the role of negative information.

We have recently been able to characterize the supported model classes of general logic programs, such as were studied in this paper. We have also obtained some further results concerning logic programs with classical negation.

Acknowledgements

The results in this paper form a portion of the author's Ph.D. thesis at the University of Michigan, written under the direction of Professor Andreas Blass, to whom the author expresses warmest appreciation. The author would also like to acknowledge useful discussions with Professor David Kueker of the University of Maryland concerning definability of minimal models. The author is especially grateful to Professor Victor Marek of the University of Kentucky for communicating his insight into these subjects.

References

[1] A. Batarekh, Topological aspects of logic programming, Ph.D. Thesis, Department of Computer Science, Syracuse University (1989).

[2] P.M. Dung and K. Kanchanasut, A fixpoint approach to declarative semantics of logic programs, *Proceedings of the North American Conference on Logic Programming* (MIT Press, Cambridge, MA, 1989) pp. 604–625.

[3] M. Gelfond and V. Lifschitz, The stable semantics for logic programs, *Proceedings of the 5th International Symposium on Logic Programming* (MIT Press, Cambridge, MA, 1988) pp. 1070–1080.

[4] W. Marek, A. Nerode, and J.C. Remmel, A theory of nonmonotonic rule systems II, Ann. of Math. and Artificial Intelligence 5 (1992) 229–264.

[5] W. Marek, A. Nerode, and J.C. Remmel, How complicated is the set of stable models of a recursive logic program? Ann. Pure and Appl. Logic 56 (1992).

[6] W. Marek, A. Nerode, and J.C. Remmel, The stable models of a predicate logic program, *Joint International Conference and Symposium on Logic Programming*, Washington, D.C., 1992.