A Connection between the Cantor-Bendixson Derivative and the Well-Founded Semantics of Logic Programs

Douglas Cenzer
Department of Mathematics,
University of Florida,
Gainesville, FL 32611-8105

Jeffrey B. Remmel
Department of Mathematics,
University of California, San Diego,
La Jolla, CA 92093

Abstract

We show that there is a close connection between the construction of the perfect kernel of a Π^0_1 class via the interation of the Cantor-Bendixson derivative through the ordinals and the construction of the well-founded semantics for logic programs via Van Gelder's alternating fixpoint construction. This connection allows us to transfer known complexity results for the perfect kernel of Π^0_1 classes to to give new complexity results for various questions about the well-founded semantics $wfs(P)$ of a logic program P.

1 Introduction

In this paper we shall study the complexity of the well-founded semantics of infinite propositional logic programs. The well-founded semantics was introduced by Van Gelder, Ross, and Schlipf (Van-Gelder, Ross, & Schlipf 1991). It provides a 3-valued interpretation to logic programs with negation and it can be viewed as an approximation to the stable semantics as defined by Gelfond-Lifschitz (Gelfond & Lifschitz 1988), see (Van-Gelder, Ross, & Schlipf 1991) and (Fitting 2002). The stable model semantics is defined by means of a fixpoint of an anti-monotone operator often denoted by $GL_P(\cdot)$. Van Gelder (Van-Gelder 1989) showed that the well-founded semantics can be defined as the alternating fixpoint of GL_P. The relationship between the well-founded semantics and inductive definition was studied by Denecker and his collaborators (Denecker 1998; Denecker, Bruynooghe, & Marek 2001).

It is known that the well-founded semantics for a finite propositional logic program can be computed in polynomial time (Van-Gelder 1989) while the problem of deciding whether a finite propositional program has a stable model is NP-complete (Marek & Truszczyński 1991). The basic results for the complexity of the well-founded semantics of infinite logic programs can be found in (Schlipf 1992) and (Fitting 2002). Complexity results for the stable model semantics of infinite logic programs can be found in (Marek, Nerode, & Remmel 1994). Basically, the both the well-founded semantics and the stable logic semantics for recursive logic programs can capture any Π^1_1 set. For example, there are recursive programs for which the well-founded semantics is Π^1_1-complete set (Schlipf 1992) and the problem of deciding whether a recursive program has a stable model is Σ^1_1-complete (Marek, Nerode, & Remmel 1994).

In what follows, we will assume that the atoms appearing in all programs come from a fixed countable set of atoms $p_0, p_1, \ldots$. We shall implicitly identify p_i with the integer i so that we can think of the atoms appearing in our programs as integers. This will allow us to give precise definitions of recursive and recursively enumerable programs.

In this paper, we shall develop a number of index set results for the well-founded semantics for logic programs. Index set results provide for a finer classification of the complexity of various decision problems. For example, suppose that ϕ_e is the partial recursive function computed by the e-th Turing machine and W_e is domain of ϕ_e. Thus $\phi_0, \phi_1, \ldots$ is a list of all partial recursive functions and $W_0, W_1, \ldots$ is a list of all recursively enumerable (r.e.) sets. It is well known that there is no uniform procedure which given e will decide whether W_e is non-empty, finite, or recursive. However, the complexity of deciding whether a given r.e. set W_e is non-empty, finite, or recursive are not the same. That is, let $Non = \{e : W_e \text{ is non-empty}\}$, $Fin = \{e : W_e \text{ is finite}\}$, and $Rec = \{e : W_e \text{ is recursive}\}$. Then it is well-known that Non is Σ^0_1-complete, Fin is Σ^0_2-complete, and Rec is Σ^0_3-complete; see (Soare 1987).

We shall be interested in the index sets associated with various properties of the well-founded semantics of recursively enumerable logic programs. That is, if $LP_0, LP_1, \ldots$ is an effective list of all recursively enumerable propositional logic programs whose set of atoms is contained in the natural numbers ω and $\mathcal{R}$ is some property of the well-founded semantics, then we are interested in classifying the complexity of the set of all e such that the well-founded semantics of LP_e had property $\mathcal{R}$. We shall show that there is a close connection between the well-founded semantics of recursively enumerable logic programs and the Cantor-Bendixson derivatives of Π^0_1 classes contained in 2^ω. In particular, we shall show that for each primitive recursive binary tree T_e, there is a recursively enumerable logic program P_e such that if λ is either a limit ordinal or 0 and α is finite, then the complexity of the $\lambda + 2\alpha$-th level of the Van Gelder alternating fixed point construction of the well-founded semantics of P_e is equivalent to the complexity of the $\lambda+\alpha$-th derivative of the Π^0_1 class of all infinite paths through T_e, $[T_e]$. Moreover, it will be case that if $\lambda + n$ is the ordinal at which the iteration of the Cantor-Bendixson derivative applied to $[T_e]$ reaches the perfect kernel $K([T_e])$, then the Van Gelder alternating fixed point of construction applied to P_e will give the well-founded semantics of P_e at level $\lambda + 2n$. Now there are many results in the literature about the complexity of index sets associated with the construction of the perfect kernel of Π^0_1 classes. Our correspondence $T_e \rightarrow P_e$ allows us to transfer such complexity results to produce new complexity results for the well-founded semantics of r.e. programs. For example, we can show that the set of all e such that the true sentences under the wff-semantics of LP_e is recursive or Δ^1_1 is a Π^1_1-complete set. Thus the problem of deciding whether the well-founded semantics of an r.e. program is recursive is a Π^1_1 complete problem. We also prove some index set results for properties that imply the well-founded semantics is relatively simple. For example, we show that the set of e such that the true sentences under the wff-semantics of LP_e is empty is Π^0_1 complete, the set of e such the false sentences under the wff-semantics of LP_e is empty is Π^0_3 complete, and the set of e such that the true sentences under the wff-semantics of LP_e is just the least model of the Horn part of the program is Π^0_2 complete.

The outline of this paper is as follows. In section 2, we shall provide the basic definitions from logic programming and recursion theory that we will need to state our results. In section 3, we shall give our correspondence between the well-founded semantics and the Cantor-Bendixson derivative of Π^0_1 classes. In section 4, we shall derive index set results for logic programs for which the well-founded semantics is especially simple.

2 Basic Definitions

In this section, we shall provide the basic definitions of the stable and well founded semantics as well as give precise definitions of recursive and recursively enumerable (r.e.) programs. We shall also give some basic definitions from recursion theory and state some key complexity results due to Cenzer and Remmel (Cenzer & Remmel 1998a) which will be used to prove our main results.

2.1 Definitions of Stable and Well-founded Semantics

A logic programming clause is a construct of the form

$$C = p \leftarrow q_1, \ldots, q_m, \neg\, r_1, \ldots, \neg\, r_n.$$

The literals $q_1, \ldots, q_m, \neg r_1, \ldots, \neg\, r_n$ form the *body* of C and the atom p is called the *head* of C. A logic program P is a set of logic programming clauses. We say that a set of atoms M is a model of a clause C if either M does not satisfy the body of C or M satisfies the head of C. M is said to be a model of a logic program P if M is a model of each of the clauses of P. Clauses C where $n = 0$ are called *Horn* clauses. P is said to be a *Horn program* if all its clauses are Horn clauses. A Horn program P always has a least model $LM(P)$. It is constructed by iterating the one-step provability operator T_P for P. That is, given a set I of atoms, we let $T_P(I)$ be equal to

$$\{p : \exists C = p \leftarrow a_1, \ldots, a_n \in P : a_1, \ldots, a_n \in I\}.$$

Then the least model of P, $LM(P)$, equals

$$T_P(\emptyset) \uparrow_\omega = \bigcup_{n \geq 1} T^n_P(\emptyset).$$

Next assume P is a logic program with negated atoms in the body of some of its clauses. Then following (Gelfond & Lifschitz 1988), we define the *stable models* of P as follows. Assume M is a collection of atoms. The *Gelfond-Lifschitz reduct* of P by M is a Horn program arising from P by first eliminating those clauses in P which contain $\neg\, r$ with $r \in M$. In the remaining clauses, we drop all negative literals from the body. The resulting program $GL_M(P)$ is a Horn program. We call M a *stable model* of P if M is the least model of $GL_M(P)$. For a Horn program P, there is a unique stable model, namely, the least model of P.

Assume that we are given a logic program P and the set of atoms of P is contained in N. We let $\mathcal{P}(N)$ denote the set of all subsets of N and for any set $M \subseteq N$, let $\overline{M} = N - M$. Then we define the operator $A_P : \mathcal{P}(N) \rightarrow \mathcal{P}(N)$ by

$$A_P(M) = LM(GL_M(P)).$$

It is well known that A_P is anti-monotone, i.e., $S \subseteq T$ implies $A_P(T) \subseteq A_P(S)$. Thus the operator $U_P = A^2_P$ is monotone. Also the operator V_P defined by

$$V_P(M) = \overline{U_P(\overline{M})}$$

is monotone. Next we define U^α_P and V^α_P for any ordinal α by

$$U^0_p(M) = M, V^0_p(M) = M,$$
$$U^{\alpha+1}_p(M) = U_P(U^\alpha_P(M)), V^{\alpha+1}_p(M) = V_P(V^\alpha_P(M))$$
$$U^\lambda_p(M) = \bigcup_{\alpha<\lambda} U^\alpha_P(M)), V^\lambda_p(M) = \bigcup_{\alpha<\lambda} V^\alpha_P(M))$$

for λ a limit ordinal.

It follows from the Knaster-Tarski Theorem (Tarski 1955) that both U_P and V_P must have least fixed points. Then we can define the set atoms that are true under the well-founded semantics to be $T_{wfs}(P) = lpf(U_P)$ and the set of atoms which are false under the well-founded semantics to be $F_{wfs}(P) = lfp(V_P)$. It is also not difficult to see that $F_{wfs}(P) = \overline{A_P(T_{wfs}(P))}$.

Van Gelder (Van-Gelder 1989) gave the following alternating fixed point algorithm to compute the well-founded semantics.

Algorithm $F_0(P) := \emptyset$ and
$T_0(P) := A_P(\overline{F_0}) = LM(GL_\omega(P))$.

$F_{\alpha+1}(P) = \overline{T_\alpha}$ and
$T_{\alpha+1}(P) = A_P(\overline{F_{\alpha+1}(P)}) = LM(GL_{\overline{F_{\alpha+1}(P)}}(P))$.

For λ a limit ordinal,
$F_\lambda(P) = \bigcup_{\alpha<\lambda} F_\alpha(P)$ and
$T_\lambda(P) = A_P(\overline{F_\lambda(P)}) = LM(GL_{\overline{F_\lambda(P)}}(P))$.

Then $F_{wfs}(P) = F_\alpha(P)$ and $T_{wfs} = T_\alpha(P)$ where α is the least ordinal such that $F_\alpha(P) = F_{\alpha+1}(P)$.

We will be most interested in the "even" stages of the alternating fixed point construction. Note that it is easy to see that for all α,

$$\begin{aligned} F_{\alpha+2}(P) &= \overline{T_{\alpha+1}(P)} = \overline{A_P(\overline{F_{\alpha+1}(P)})} \\ &= \overline{A_P(T_\alpha(P))} = \overline{A_P(A_P((\overline{F_\alpha}(P)))} \\ &= V_P(F_\alpha(P)) \text{ and} \\ T_{\alpha+2}(P) &= A_P(\overline{F_{\alpha+2}(P)}) = A_P(T_{\alpha+1}(P)) \\ &= A_P(A_P(\overline{F_{\alpha+1}(P)})) = A_P(A_P(T_\alpha(P)) \\ &= U_P(T_\alpha(P)). \end{aligned}$$

Thus for n finite and λ a limit ordinal, $F_{2n}(P) = V_P^n(\emptyset)$, $F_\lambda(P) = V_P^\lambda(\emptyset)$, and $F_{\lambda+2n}(P) = V_P^{\lambda+n}(\emptyset)$. Similarly, $T_{2n}(P) = U_P^n(T_0(P))$, $T_\lambda(P) = U_P^\lambda(T_0(P))$, and $T_{\lambda+2n}(P) = U_P^{\lambda+n}(T_0(P))$.

2.2 Basic Definitions from Recursion Theory

Let $\omega = \{0, 1, 2, \ldots\}$ denote the set of natural numbers, let $\omega^{<\omega}$ denote the set of all finite sequences from ω and let $2^{<\omega}$ denote the set of all finite sequences of 0's and 1's. Let $c(x_1, x_2, \ldots, x_n)$ be the standard coding of finite sequences into natural numbers as in (Cenzer & Remmel 1998b).

We shall assume that the atoms of the programs that we consider are contained in the natural numbers. Thus given a program clause

$$C = p \leftarrow q_1, \ldots, q_m, \neg\, r_1, \ldots, \neg\, r_n \tag{1}$$

we define the code of C, $code(C)$, to be $c(m, n, p, c(q_1, \ldots, q_m), c(r_1, \ldots, r_n))$. Then we say that a program P consisting of clauses of the form of (1) is recursive (r.e., etc.) if the set of codes of its clauses is recursive (r.e., etc.). We let $Horn(P)$ denote the set of Horn clauses of P. Thus if P is recursive (r.e.), then $Horn(P)$ will be recursive (r.e.). Let ϕ_e denote the partial recursive function computed by e-th Turing machine M_e so that $\phi_0, \phi_1, \ldots$ is an effective enumeration of all partial recursive functions. We define the e-th r.e. set W_e to be the domain of ϕ_e so that $W_0, W_1, \ldots$ is an effective enumeration of all r.e. sets. We let LP_e denote the set of all x in W_e such that x is a code of a clause of the form of (1). Thus $LP_0, LP_1, \ldots$ is an effective enumeration of all r.e. logic programs P whose set of atoms is a subset ω.

Given $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_k)$ in $\omega^{<\omega}$, we write $\alpha \sqsubseteq \beta$ if α is initial segment of β, that is, if $n \leq k$ and $\alpha_i = \beta_i$ for $i \leq n$. For any finite sequence $\sigma \in \{0,1\}^*$, let $I[\sigma] = \{x \in \{0,1\}^\omega : \sigma \sqsubseteq x\}$. For the rest of this paper, we identify a finite sequence $\alpha = (\alpha_1, \ldots, \alpha_n)$ with its code $c(\alpha)$. We let 0 be the code of the empty sequence $\emptyset$. Thus, when we say a set $S \subseteq \omega^{<\omega}$ is recursive, r.e., etc., we mean the set $\{c(\alpha) \colon \alpha \in S\}$ is recursive, r.e., etc. A *tree* T is a nonempty subset of $\omega^{<\omega}$ such that T is closed under initial segments. A function $f \colon \omega \to \omega$ is an infinite *path* through T if for all n, $(f(0), \ldots, f(n)) \in T$. We let $[T]$ denote the set of all infinite paths through T. A set A of functions is a Π^0_1-class if there is a recursive predicate R such that $A = \{f \colon \omega \to \omega : \forall_n (R((f(0), \ldots, f(n)))\}$. It is not difficult to see that if A is a Π^0_1-class, then $A = [T]$ for some recursive tree $T \subseteq \omega^{<\omega}$.

One of the goals of this paper is develop a close connection between the well-founded semantics and Cantor-Bendixson derivatives on closed sets Q contained in 2^ω. The Cantor-Bendixson derivative $D(Q)$ is defined to be the set of nonisolated members of Q. The perfect kernel $K(Q)$ is defined to be the (possibly empty) largest perfect subset of Q. Thus $K(Q)$ is empty if and only if Q is countable. $K(Q)$ may be obtained by iterating the derivative through the recursive ordinals, where $D^{\alpha+1}(Q) = D(D^\alpha(Q))$ and $D^\lambda(Q) = \bigcap_{\alpha<\lambda} D^\alpha(Q)$ for limit ordinals Q. Then $K(Q) = \bigcap_\alpha D^\alpha(Q)$, where the intersection ranges over all ordinals. The Cantor-Bendixson rank $rk(Q)$ is the least ordinal α such that $D^\alpha(Q) = K(Q)$. For a Π^0_1 class Q, it is known that $rk(Q) \leq \omega_1^{C\text{-}K}$, the least nonrecursive ordinal.

To establish our connection between the well-founded semantics and the Cantor-Bendixson derivative, we consider index sets for strong Π^0_n binary classes and also index sets for the cardinality of the Cantor-Bendixson derivatives. We only consider binary classes. These problems were first studied in the context of Polish spaces by Kuratowski, see (Kuratowski 1970), where the Cantor-Bendixson derivative is viewed as a mapping from the space of compact subsets of $\{0,1\}^\omega$ to itself. Kuratowski showed that the derivative is a Borel map of class exactly two. In particular, he showed that the family $D^{-1}(\{\emptyset\})$ of finite closed sets is a universal $\mathbf{\Sigma^0_2}$ class and posed the problem of determining the exact Borel class of the iterated operator D^α. Cenzer and Mauldin showed in (Cenzer & Mauldin 1982) and that the iterated operator D^n is of Borel class exactly $2n$ for finite n and that for any limit ordinal λ and any finite n, $D^{\lambda+n}$ is of Borel class exactly $\lambda + 2n + 1$. In particular it is shown that for any α, the family T_α of closed sets K such that $D^\alpha(K) = \emptyset$ is a universal $\mathbf{\Sigma^0_{2\alpha}}$ set. Lempp gave effective versions of this result in (Lempp 1987).

Let $T_0, T_1, \ldots$ be an effective list of all primitive recursive tree contained in $\{0,1\}$. Then it is well known that $[T_0], [T_1], \ldots$ is an effective list of all Π^0_1 classes. For any fixed set X, we let $[T_0^X], [T_1^X], \ldots$ enumerate the binary classes which are Π^0_1 in X. That is, let π_e^X be the e-th function primitive recursive in X and

$$T_e^X = \{\emptyset\} \cup \{\sigma : (\forall \tau \prec \sigma) \pi_e^X(\langle \tau \rangle) = 1\}.$$

For any property $\mathcal{R}$, let $I_P^X(\mathcal{R}) = \{e : [T_e^X] \text{ satisfies } \mathcal{R}\}$.

The following result was proved by Cenzer and Remmel (Cenzer & Remmel 1998a).

Theorem 2.1. *For any set X,*

1. $I_P^X(\text{empty})$ *is* $\Sigma^{0\,X}_1$ *complete,*
2. $I_P^X(=1)$ *is* $\Pi^{0\,X}_2$ *complete.*
3. *For any integer* $c > 0$, $I_P^X(> c)$ *is* $\Sigma^{0\,X}_2$ *complete and* $I_P^X(= c+1)$ *is* $D^{0\,X}_2$ *complete.*
4. $I_P^X(\text{finite})$ *is* $\Sigma^{0\,X}_3$ *complete.* □

The Σ^0_α sets may be defined for any recursive ordinal α and the strong $\Pi^0_{\alpha+1}$ classes may be defined as set of infinite paths through a Σ^0_α tree. To classify index sets connected with the transfinite Cantor-Bendixson derivatives of Π^0_1 classes, Cenzer and Remmel (Cenzer & Remmel 1998a) established a correspondence between the $\Pi^0_{2\alpha+1}$ classes and the α-th Cantor-Bendixson derivatives of Π^0_1 classes. When $\alpha = \lambda + n$ for a limit ordinal λ and finite n, define $2\alpha = \lambda + 2n$, $2\alpha + 1 = \lambda + 2n + 1$, and $2\lambda - 1 = \lambda$.

Theorem 2.2. *For any recursive ordinal α and any r. b. $\Pi^0_{2\alpha+1}$ class Q, there exists a Π^0_1 class P of sets and a homeomorphism H from Q onto $D^\alpha(P)$ such that $x \leq_T H(x) \leq x \oplus 0^{2\alpha-1}$ for all $x \in Q$.*

Cenzer and Remmel (Cenzer & Remmel 1998a) proved the following.

Theorem 2.3. *For any computable ordinal α,*

1. *$\{e : D^\alpha([T_e])$ is empty$\}$ is $\Sigma^0_{2\alpha+1}$ complete and $\{e : D^\alpha([T_e])$ is nonempty$\}$ is $\Pi^0_{2\alpha+1}$ complete.*
2. *$\{e : card(D^\alpha([T_e])) = 1\}$ is $\Pi^0_{2\alpha+1}$ complete.*
3. *For any positive integer c, $\{e : card(D^\alpha([T_e])) \leq c\}$ is $\Sigma^0_{2\alpha+2}$ complete and $\{e : card(D^\alpha([T_e])) > c\}$ is $\Pi^0_{2\alpha+2}$ complete.*
4. *$\{e : D^\alpha([T_e])$ is infinite$\}$ is $\Pi^0_{2\alpha+3}$ complete and $\{e : D^\alpha([T_e])$ is finite$\}$ is $\Sigma^0_{2\alpha+3}$ complete.*

Theorem 2.4. *The following index sets are all Π^1_1 complete:*

1. *$\{e : K[T_e]$ is countable$\} = \{e : K([T_e])$ is empty$\}$.*
2. *$\{e : K([T_e])$ is $\Delta^1_1\} = \{e : K([T_e])$ is $\Pi^1_1\}$.*
3. *$\{e : K([T_e])$ is recursive$\}$.*

3 The Cantor-Bendixson Derivative and the Well-Founded Semantics

In this section, we shall define a simple logic program P_e for each primitive recursive tree T_e such that for any ordinal λ which is either a limit ordinal or 0 and any finite n, $T_{\lambda+2n}(P_e) = \{\sigma \in 2^{<\omega} : I[\sigma] \cap D^{\lambda+n([T_e])} = \emptyset\}$. This shows that there is a simple connection between the construction of a perfect kernel of Π^0_1 classes and Van Gelder's alternating fixed point construction of the well-founded semantics of r.e. programs. We shall then use the correspondence $T_e \to P_e$ to derive some new index set results for the well-founded semantics by transferring the index set results given in section 2.

Recall that we have an effective enumeration $LP_0, LP_1, \ldots$ of all r.e. logic programs whose set of atoms is contained in the natural numbers ω. For any property $\mathcal{R}$ of logic programs, we let

$$I_{LP}(\mathcal{R}) = \{e : LP_e \text{ has property } \mathcal{R}\}. \tag{2}$$

We shall construct a recursive logic program P_e depending on T_e. For each string $\sigma \in \{0,1\}^*$, we will identify any σ which occurs in P_e with 2 times its code $c(\sigma)$ and any σ^* which occurs in P_e with $2c(\sigma) + 1$. It will follow for our construction that the atoms of P_e will be the set of all σ and σ^* with $\sigma \in \{0,1\}^*$ so that the we can identify the Herbrand base of P_e with the set of natural numbers ω. This given, we define the logic program P_e to consist of the following set of clauses.

(i) $\sigma \leftarrow$ for $\sigma \notin T_e$

(ii) $\sigma \leftarrow \sigma^\frown 0, \sigma^\frown 1$ for all σ

(iii) $\sigma^* \leftarrow \neg\sigma^\frown\tau^\frown 0, \neg\sigma^\frown\tau^\frown 1$ for all σ and τ

(iv) $\sigma \leftarrow \neg\sigma^*$ for all σ.

The intended model here is $M = \{\sigma : I(\sigma) \cap K([T_e]) = \emptyset\} \cup \{\sigma^* : I[\sigma] \cap K([T_e]) \neq \emptyset\}$. We claim that M is a stable model of P_e. Clearly, $GL_M(P_e)$ has the rules

(i) $\sigma \leftarrow$ for $\sigma \notin T_e$

(ii) $\sigma \leftarrow \sigma^\frown 0, \sigma^\frown 1$ for all σ

(iii) $\sigma^* \leftarrow$ for all σ such that there exists a τ such that $\sigma^\frown\tau^\frown 0$ and $\sigma^\frown\tau^\frown 1$ are both not in M

(iv) $\sigma \leftarrow$ for all σ such that $\sigma^* \notin M$.

If $\sigma \in M$, then $\sigma^* \notin M$ so that $\sigma \in LM(GL_M(P_e)$ by rule (iv). If $\sigma^* \in M$, then σ has an infinite extension $x \in K([T_e])$. Thus since $K([T_e])$ is perfect, there exists τ such that both $\sigma^\frown\tau^\frown 0$ and $\sigma^\frown\tau^\frown 1$ both have infinite extensions in $K([T_e])$. It follows that both $\sigma^\frown\tau^\frown 0$ and $\sigma^\frown\tau^\frown 1$ are not in M, so that $\sigma^* \in LM(GL_M(P_e))$ by clause (iii).

On the other hand, if $\sigma \in LM(GL_M(P_e))$, then we can argue by induction on the length of the proof scheme for σ that $\sigma \in M$. If σ comes in by clause (i), then $\sigma \notin T_e$, so certainly $\sigma \in M$. If σ comes in by clause (ii), then by induction both $\sigma^\frown 0$ and $\sigma^\frown 1$ are in M, so that

$$I[\sigma] \cap K([T_e]) = \\ (I[\sigma^\frown 0] \cap K([T_e])) \cup (I[\sigma^\frown 1] \cap K([T_e]) = \emptyset,$$

and therefore $\sigma \in M$. If σ comes in by clause (iv), then $\sigma^* \notin M$, so that $\sigma \in M$.

If $\sigma^* \in LM(GL_M(P_e))$, then some $\sigma^\frown\tau^\frown 0 \notin M$, so that $I[\sigma] \cap K([T_e]) \supset I[\sigma^\frown\tau^\frown 0] \cap K([T_e]) \neq \emptyset$ and thus $\sigma^* \in M$.

The main result of this paper is the following.

Theorem 3.1. *For all e and for all recursive ordinals α of the form $\lambda + 2n$ where n is finite and λ is either a limit ordinal or 0,*

$$T_{\lambda+2n}(P_e) = \{\sigma : I[\sigma] \cap D^{\lambda+n}([T_e]) = \emptyset\}, \tag{3}$$

$$F_{\lambda+2n}(P_e) = \{\sigma^* : card(I[\sigma] \cap D^{\lambda+n-1}([T_e])) \leq 1\} \tag{4}$$

if $n > 0$, and

$$F_\lambda(P_e) = \{\sigma^* : card(I[\sigma] \cap D^\lambda([T_e])) = \emptyset\} \tag{5}$$

if λ is a limit ordinal. Hence

$$\begin{aligned} T_{wfs}(P_e) &= \cup_{\lambda+2n} T_{\lambda+2n}(P_e) \\ &= \{\sigma : I[\sigma] \cap K([T_e]) = \emptyset\}. \end{aligned}$$

Proof. Consider the levels of $F_\alpha(P_e)$ and $T_\alpha(P_e)$. $GL_{\omega-F_0}(P_e)$ has only clauses (i) and (ii). Now if $I[\sigma] \cap [T_e] = \emptyset$, then by Konig's Lemma, the set of $\tau \in T_e$ which extend σ is finite so that we will be able to derive σ by repeated use of the clauses in (i) and (ii). Thus

$$T_0(P_e) = \{\sigma : I[\sigma] \cap [T_e] = \emptyset\}.$$

Then $GL_{T_0(P_e)}(P_e)$ has clauses (i) and (ii) together with two families of clauses. First there are $\sigma^* \leftarrow$ for all σ such that for some τ both $I[\sigma^\frown\tau^\frown 0]$ and $I[\sigma^\frown\tau^\frown 1]$ meet $[T_e]$, that is, if $card(I[\sigma] \cap [T_e]) \geq 2$. Finally there are clauses $\sigma \leftarrow$ for all σ such that $\sigma^* \notin T_0(P_e)$, which is to say for all σ. Thus

$$T_1(P_e) = LM(GL_{T_0(P_e)}(P_e)) =$$
$$\{0,1\}^* \cup \{\sigma^* : card(I[\sigma] \cap [T_e]) \geq 2\}.$$

This means that

$$F_2(P_e) = \omega - LM(GL_{T_0(P_e)}(P_e)) =$$
$$\{\sigma^* : card(I[\sigma] \cap [T_e]) \leq 1\}.$$

This establishes the base case.

Next observe that the limit case follows immediately by induction and compactness. So assume that λ is a limit ordinal and $F_\lambda(P_e) = \{\sigma^* : I[\sigma] \cap D^\lambda([T_e]) = \emptyset\}$. Then $\overline{F_\lambda(P_e)} = \{0,1\}^* \cup \{\sigma^* : card(I[\sigma] \cap D^\lambda([T_e])) \neq \emptyset\}$. Hence $GL_{\overline{F_\lambda(P_e)}}(P_e)$ consists of clauses (i) and (ii) plus the set of all $\sigma \leftarrow$ such that $I[\sigma] \cap D^\lambda([T_e]) = \emptyset$. But it is easy to see that if both $I[\sigma^\frown 0] \cap D^\lambda([T_e]) = \emptyset$ and $I[\sigma^\frown 1] \cap D^\lambda([T_e]) = \emptyset$, then $I[\sigma] \cap D^\lambda([T_e]) = \emptyset$ so that

$$T_\lambda(P_e) = LM(GL_{\overline{F_\lambda(P_e)}}(P_e)) =$$
$$\{\sigma : I[\sigma] \cap D^\lambda([T_e]) = \emptyset\}.$$

Then we can reason exactly as in the base case to conclude that $F_{\lambda+2}(P_e) = \{\sigma^* : card(I[\sigma] \cap D^\lambda([T_e]) \leq 1\}$.

Finally suppose that for any $n \geq 1$,

$$F_{\lambda+2n}(P_e) =$$
$$\{\sigma^* : card(I[\sigma] \cap D^{\lambda+n-1}([T_e])) \leq 1\}.$$

Then $GL_{\overline{F_{\lambda+2n}(P_e)}}(P_e)$ has the rules (i) and (ii) for all σ, and has the rules $\sigma \leftarrow$ for all σ such that $I[\sigma] \cap D^{\lambda+n-1}([T_e]) \leq 1\}$. It follows that

$$T_{\lambda+2n}(P_e) = LM(GL_{\overline{F_{\lambda+2n}}}) =$$
$$\{\sigma : I[\sigma] \cap D^{\lambda+n-1}([T_e]) \text{ is finite}\},$$

which equals

$$\{\sigma : I[\sigma] \cap D^{\lambda+n}([T_e]) \text{ is empty}\}$$

as desired.

Given $T_{\lambda+2n}(P_e) = \{\sigma : I[\sigma] \cap D^{\lambda+n}([T_e]) = \emptyset\}$, we see that $GL_{T_{\lambda+2n}(P_e)}(P_e)$ has the rules $\sigma \leftarrow$ for all σ, and the rules $\sigma^* \leftarrow$ for all σ such that there exists a τ such that both $I[\sigma^\frown\tau^\frown 0]$ and $I[\sigma^\frown\tau^\frown 1]$ meet $D^{\lambda+n}([T_e])$, which is to say that $card(I[\sigma] \cap D^{\lambda+n}([T_e]) \geq 2$. Thus

$$LM(GL_{T_{\lambda+2n}(P_e)}(P_e)) =$$
$$\{0,1\}^* \cup \{\sigma^* : card(I[\sigma] \cap D^{\lambda+n}([T_e])) \geq 2\}.$$

Hence

$$F_{\lambda+2n+2} = \omega - LM(GL_{T_\alpha(P_e)}(P_e)) =$$
$$\{\sigma^* : card(I[\sigma] \cap D^{\lambda+n}([T_e])) \leq 1\},$$

as desired. It now follows that

$$T_{wfs}(P_e) = \cup_\alpha T_\alpha = \{\sigma : I[\sigma] \cap K([T_e]) = \emptyset\}$$

and hence *every* stable model of P includes $\{\sigma : I[\sigma] \cap K(Q) = \emptyset\}$. □

Before stating our index set results, some more background is needed from computability theory (Cenzer & Remmel ; Soare 1987). For an infinite limit ordinal λ and natural number n, our convention is that $2(\lambda+n) = \lambda+2n$, so that $2(\lambda+n)+1 = \lambda+2n+1$. For a limit ordinal λ, a set A is Σ^0_λ if it is the effective union $A = \cup_n A_n$ of sets such that each A_n is Σ^0_α for some $\alpha < \lambda$ and a set is Π^0_λ if its complement is Σ^0_λ. A set $A \subseteq \omega$ is said to be Σ^0_α *complete* if it is Σ^0_α and for any Σ^0_α set B, there is a computable function φ such that, for any n, $n \in B \iff \phi(n) \in A$. Similar definitions apply for Π^0_α and also for $\Sigma^1_1, = \Pi^1_1$ and other notions of definability.

Theorem 3.2. *Let $T_{e,\alpha} = T_\alpha(LP_e)$ and $F_{e,\alpha} = F_\alpha(LP_e)$ be the sequence of sets defined in the alternating fixpoint algorithm to compute the well-founded semantics for LP_e. Then for any recursive ordinal α,*

(i) If α is finite, then $\{\langle e,a\rangle : a \in F_{e,\alpha}\}$ is Π^0_α and $\{\langle e,a\rangle : a \in T_{e,\alpha}\}$ is $\Sigma^0_{\alpha+1}$.

(ii) If α is a limit ordinal, then $\{\langle e,a\rangle : a \in F_{e,\alpha}\}$ and $\{\langle e,a\rangle : a \in T_{e,\alpha}\}$ are both Σ^0_α.

(iii) If $\alpha = \lambda+n$ where λ is a limit ordinal and $n > 0$ is a natural number, then $\{\langle e,a\rangle : a \in F_{e,\alpha}\}$ is $\Pi^0_{\lambda+n-1}$ and $\{\langle e,a\rangle : a \in T_{e,\alpha}\}$ is $\Sigma^0_{\lambda+n}$.

Proof. First observe that

$$B_{LP_e}(M) = LM(GL_{\omega-M}(LP_e))$$

is Σ^0_1 in M so that

$$V_{LP_e}(M) = \omega - B_{LP_e}(M).$$

is Π^0_1 in M.

But then

$$F_\alpha(LP_e) = V^\alpha_{LP_e}(\emptyset),$$

so that the complexity of F_α now follows from standard inductive definability results (Hinman 1978). In particular, since $F_\lambda(LP_e) = \bigcup_{\alpha<\lambda} F_\alpha(LP_e)$ is Σ^0_λ and Γ is monotone, then $F_{\lambda+1}(LP_e)$ is $\Pi^0_{\lambda+1}$.

Then for any ordinal α, the complexity of $T_\alpha(LP_e)$ follows from $T_\alpha(LP_e) = LM(GL_{\omega-F_\alpha(LP_e)}(LP_e))$. In particular, $T_{\lambda+1}(LP_e)$ is $\Sigma^0_{\lambda+2}(LP_e)$. □

In fact, one can use results like Theorem 2.4 to show that these complexities are exact, that is, each index set is in fact complete for its level of complexity.

Next we can apply Theorem 2.4 and Theorem 3.1 to derive the following index set results for the well-founded semantics.

Theorem 3.3. *Let R be any infinite and coinfinite recursive set. Then the following index sets are all Π^1_1 complete:*

(i) $\{e : T_{wfs}(LP_e)$ *is recursive*$\}$

(ii) $\{e : R \subseteq T_{wfs}(LP_e)\}$, *and*

(iii) $\{e : T_{wfs}(LP_e)$ *is* $\Delta^1_1\}$.

Proof. The upper bound on the complexity follows from the fact that $T_{wfs}(LP_e)$ can be obtained from the closure of a Π^0_2 monotone inductive operator. Therefore $T_{wfs}(LP_e)$ is Δ^1_1 if and only if there exists a countable α such that the inductive operator closes at stage α and, hence, $T_\alpha(LP_e) = T_{\alpha+1}(LP_e)$ and $F_\alpha(LP_e) = F_{\alpha+1}(LP_e)$. This is a Π^1_1 condition by the Stage Comparison Theorem (Hinman 1978), p. 105.

It follows from the proof of Theorem 3.1 that there is a 1:1 recursive function f such that $P_e = LP_{f(e)}$. Since

$$T_{wfs}(LP_{f(e)}) = \{\sigma \in \{0,1\}^* : I[\sigma] \cap K([T_e]) = \emptyset\},$$

it is easy to see that $K([T_e])$ is recursive (Δ^1_1) if and only if $T_{wfs}(LP_{f(e)})$ is recursive (Δ^1_1). Hence f shows that $\{e : K([T_e])$ is recursive$\}$ is 1:1 reducible to $\{e : T_{wfs}(LP_e)$ is recursive$\}$ and $\{e : K([T_e])$ is $\Delta^1_1\}$ is 1:1 reducible to $\{e : T_{wfs}(LP_e)$ is $\Delta^1_1\}$. Thus the Π^1_1-completeness for parts (i) and (iii) follow from Theorem 2.4. For the Π^1_1-completeness of part (ii), note that $K([T_e]) = \emptyset$ if and only if $\{0,1\}^* \subseteq T_{wfs}(LP_{f(e)})$. Thus again we can use the fact that $\{e : K([T_e]) = \emptyset\}$ is Π^1_1 complete to establish the Π^1_1 completeness of part (ii) in the case where $R = \{2x : x \in \omega\}$. The Π^1_1 completeness for any other infinite-coinfinite set R can be established by simply recursively renumbering the codes of $\{\sigma, \sigma^* : \sigma \in \{0,1\}^*\}$. $\square$

4 Index sets for logic programs with simple well-founded semantics

In this section, we will derive a number of index sets results for logic programs whose well-founded semantics is extremely simple.

Our first result is to consider the property of the well-founded semantics being trivial. That is, it is always the case that $T_{wfs}(P)$ contains the least model of the Horn part of P, i.e., $LM(Horn(P)) \subseteq T_{wfs}(P)$. Thus we will say that the well-founded semantics of P is trivial if $T_{wfs}(P) = LM(Horn(P))$. Thus we are interested in the complexity of the set

$$\begin{aligned} &I_{LP}(triv\text{-}wfs) = \\ &\{e : T_{wfs}(LP_e) = LM(Horn(LP_e))\}. \end{aligned} \qquad (6)$$

Theorem 4.1. *$I_{LP}(triv\text{-}wfs)$ is Π^0_2-complete.*

Proof. Note that $T_0 = LM(G_\omega(P)) = LM(Horn(P))$ so that $F_1 = \overline{LM(Horn(P))}$ and $T_1 = LM(GL_{LM(Horn(P))}(P))$. Now $T_{wfs}(P) = LM(Horn(P))$ if and only if $T_1 = LM(Horn(P)) = LM(GL_{LM(Horn(P))}(P))$. But clearly, $Horn(P) \subseteq GL_{LM(Horn(P))}$ and hence $LM(Horn(P)) \subseteq LM(GL_{LM(Horn(P))}(P))$. Thus $T_1 = LM(Horn(P)) = LM(GL_{LM(Horn(P))}(P))$ if and only if $LM(GL_{LM(Horn(P))}(P)) \subseteq LM(Horn(P))$. But the least model of $Horn(P)$ is r.e. so that $GL_{LM(Horn(P))}(P)$ is recursive in $0'$ and $LM(GL_{LM(Horn(P))}(P))$ is r.e. in $0'$. Hence $LM(GL_{LM(Horn(P))}(P))$ is a Σ^0_2-set. Now $LM(GL_{LM(Horn(P))}(P)) \subseteq LM(Horn(P))$ if and only if for all x, either $(\neg(x \in LM(GL_{LM(Horn(P))}(P)))$ or $\vee(x \in LM(Horn(P))))$ and, hence, it is a Π^0_2 predicate. Thus $I_{LP}(triv\text{-}wfs)$ is Π^0_2.

To show that $I_{LP}(triv\text{-}wfs)$ is Π^0_2-complete, we will use that fact that $Inf = \{e : W_e$ is infinite$\}$ is a Π^0_2 complete set. For any e, we can effectively construct an effective enumeration $0 = a_0, 1 = a_1, a_2, \ldots$ of $W_e \cup \{0,1\}$ where W_e is the e-th r.e. set. We then define an r.e. program P_e which contains the clauses

(i) $2a_i \leftarrow$ for all i such that a_{i+1} exists and

(ii) $2i+1 \leftarrow \neg 2a_i$ for all i such that a_i exists.

Clearly, there is a one-to-one recursive function f such that $LP_{f(e)} = P_e$. It can be checked that W_e is infinite if and only if $T_{wfs}(LP_e) = LM(Horn(LP_e))\}$.

$\square$

Theorem 4.2. $\{e : T_{wfs}(LP_e) = \emptyset\}$ *is Π^0_1 complete.*

Proof. It is easy to see that $T_{wfs}(LP_e) = \emptyset$ if and only if $T_0(LP_e) = T_1(LP_e) = \emptyset$. But $T_0(LP_e) = LM(Horn(LP_e)) = \emptyset$ implies $F_1 = \omega$ and $T_1 = LM(GL_\emptyset(LP_e))$. But clearly $Horn(P) \subseteq GL_\emptyset(LP_e)$ so that $T_0(LP_e) = T_1(LP_e) = \emptyset$ if and only if $LM(GL_\emptyset(LP_e)) = \emptyset$. But $Gl_\emptyset(LP_e)$ is an r.e. program so that $LM(GL_\emptyset(LP_e))$ is r.e. and hence the question of whether it is empty is a Π^0_1 predicate.

To see that $\{e : T_{wfs}(LP_e) = \emptyset\}$ is Π^0_1 complete, we use the fact that $\{e : W_e = \emptyset\}$ is complete Π^0_1 set. We construct a program P_e for each r.e. set W_e as follows. We let $W_{e.s}$ denote the set of all $x \leq s$ such that $\phi_e(x)$ converges in s or fewer steps. Then P_e will consists of the following clauses:

(i) $s \leftarrow s+1$ if $W_{e,s} = \emptyset$ and

(ii) $s \leftarrow$ if $W_{e,s} \neq \emptyset$.

It is then easy to see that P_e is Horn program and that $T_1(P_e) = \emptyset$ if and only $W_e = \emptyset$. Clearly there is one-to-one recursive function such that $P_e = LP_{f(e)}$. Thus $W_e = \emptyset$ if and only if $f(e) \in \{e : T_{wfs}(LP_e) = \emptyset\}$ so that $\{e : T_{wfs}(LP_e) = \emptyset\}$ is Π^0_1 complete. $\square$

Theorem 4.3. $\{e : F_{wfs}(LP_e) = \emptyset\}$ *is Π^0_3 complete.*

Proof. First note that $F_{wfs}(LP_e) = \emptyset$ if and only if $F_2(LP_e) = \emptyset$. Since $F_2(LP_e)$ is a Π^0_2 set it follows that the predicate $F_2(LP_e) = \emptyset$ is Π^0_3.

For the completeness, we will reduce an arbitrary Π^0_3 set A to $\{e : F_{wfs}(LP_e) = \emptyset\}$. Let R be a recursive predicate such that

$$e \in A \iff (\forall m)(\exists n)(\forall p) R(e,m,n,p).$$

Define the program P_e with the following rules:

(i) $c_p \leftarrow$ for all p;
(ii) $b_{m,n} \leftarrow c_p$ for all m, n, p such that $\neg R(e, m, n, p)$;
(iii) $a_m \leftarrow \neg b_{m,n}$ for all m, n;
(iv) $b_{m,n} \leftarrow a_m$ for all m, n.

Here we assume that $\{c_p : p \in \omega\}$, $\{b_{m,n} : m, n \in \omega\}$ and $\{a_m : m \in \omega\}$ are pairwise disjoint recursive sets whose union is ω.

Then it is easy to see that

$$T_0(P_e) = \{c_p : p \in N\} \cup \{b_{m,n} : (\exists p)\neg R(e, m, n, p)\}.$$

It follows that $GL_{T_0(P_e)}(P_e)$ will have rules (i), (ii) and (iv), together with rules $a_m \leftarrow$ such that $(\exists n)(\forall p)R(e, m, n, p)$. It can be checked that $F_2(P_e) = \emptyset$ if and only if $e \in A$. $\square$

Acknowledgements

The first author was partially supported by National Science Foundation awards NSF grants DMS 0532644, 0554841 and 00062393 and the second author was partially supported by NSF grants DMS 0400507 and 0654060.

References

Cenzer, D., and Mauldin, R. D. 1982. On the borel class of the derived set operator. *Bulletin Societe Math. France* 110:357–380.

Cenzer, D., and Remmel, J. B. *Effectively Closed Sets*. ASL Lecture Notes in Logic. Association for Symbolic Logic. to appear.

Cenzer, D., and Remmel, J. B. 1998a. Index sets for π_1^0 classes. *Annals of Pure and Applied Logic* 93:3–61.

Cenzer, D., and Remmel, J. B. 1998b. π_1^0 classes in mathematics. In Y. Ersov, S. Goncharov, A. N., and Remmel, J. B., eds., *Handbook of Recursive Mathematics*, Studies in Logic Found. Math #139. Elsevier. 623–821.

Denecker, M.; Bruynooghe, M.; and Marek, V. 2001. Logic programming revisited: Logic programs as inductive definitions. *ACM Transactions on Computational Logic* 2:623–654.

Denecker, M. 1998. The well-founded semantics is the principle of inductive definition. In *Proceedings of the 6th European Workshop on Logics in Artificial Intelligence*, Lecture Notes in Artificial Intelligence #1489. Springer-Verlag. 1–16.

Fitting, M. 2002. *Fixpoint semantics for logic programming–a survey*, volume 278.

Gelfond, M., and Lifschitz, V. 1988. The stable semantics for logic programs. In Kowalski, R., and Bowen, K., eds., *Proceedings 5th Int'l. Symp. Logic Programming*. MIT Press. 1070–1080.

Hinman, P. G. 1978. *Recursion-Theoretic Hierarchies*. Perspectives in Mathematical Logic. Springer-Verlag.

Kuratowski, K. 1970. Some problems concerning semi-continuous set-valued mappings. In *Set-Valued Mappings, Selections and Topological Properties of* 2^X, Lecture Notes in Mathematics #171. Springer-Verlag. 45–48.

Lempp, S. 1987. Hyperarithmetical index sets in recursion theory. *Transactions of the American Mathematical Society* 303:559–583.

Marek, W., and Truszczyński, M. 1991. Autoepistemic logic. *J. ACM* 38:588–619.

Marek, W.; Nerode, A.; and Remmel, J. B. 1994. The stable models of a predicate logic program. *Journal of Logic Programming* 21:129–154.

Schlipf, J. 1992. The expressive power of the logic programming semantics. *J. Comp. and System Sciences* 51:64–86.

Soare, R. 1987. *Recursively Enumerable Sets and Degrees*. Springer-Verlag.

Tarski, A. 1955. A lattice-theoretical theorem and its applications. *Pacific J. Math.* 5:285–309.

Van-Gelder, A.; Ross, K.; and Schlipf, J. 1991. The well-founded semantics for general logic programs. *J. ACM* 38(3):620–650.

Van-Gelder, A. 1989. The alternating fixpoints of logic programs with negation. In *ACM Symposium on Principles of Database Systems*. 1–10.