amai manuscript No.
(will be inserted by the editor)

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

Douglas Cenzer · Jeffrey B. Remmel

Abstract Results of Schlipf [S95] and Fitting [F01] show that the well-founded semantics of a finite predicate logic program can be quite complex. In this paper, we show that there is a close connection between the construction of the perfect kernel of a Π^0_1 class via the iteration of the Cantor-Bendixson derivative through the ordinals and the construction of the well-founded semantics for finite predicate 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 give new complexity results for various questions about the well-founded semantics $wfs(P)$ of a finite predicate logic program P.

Keywords logic program · well-founded semantics · Cantor-Bendixson derivative

Mathematics Subject Classification (2000) 68T27 · 03B70 · 68N17

1 Introduction

The main goal of this paper is to show that there is a close connection between the alternating fixed point construction of the well-founded semantics of a finite predicate logic program due to Van Gelder [V89, V93] and the classical topological construction of the perfect kernal from point set topology

Cenzer was partially supported by the NSF grant DMS-652372 and Remmel by the NSF grant DMS-0654060.

D. Cenzer
University of Florida, Gainesville, FL 32611-8105, USA
Tel.: 352-392-0281
Fax: 352-392- E-mail: cenzer@ufl.edu

J.B. Remmel
University of California, San Diego, La Jolla, CA 92093, USA
E-mail: jremmel@ucsd.edu

via the iteration of the Cantor-Bendixson derivative through the ordinals. In particular, we shall show that there is a simple coding which will allows us to transfer complexity results about the perfect kernal of effectively closed sets in 2^ω to complexity results about the well-founded semantics for finite predicate logic programs. Here $\omega = \{0, 1, \ldots\}$ is the natural numbers and 2^ω is the set of all infinite sequences of 0s and 1s. The complexity of the construction of the perfect kernal of effectively closed sets in 2^ω has been extensively studied by recursion theorists. Our coding will then allow us derive new complexity results about the well-founded semantics of finite predicate logic programs by transferring known complexity results about the construction of the perfect kernal of effectively closed sets in 2^ω.

The well-founded semantics was introduced by Van Gelder, Ross, and Schlipf [VRS91]. 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 [GL88], see [VRS91] and [F01]. The stable model semantics is defined by means of a fixpoint of anti-monotone operator often denoted by $GL_P(\cdot)$. Van Gelder [V89, V93] 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 definitions was studied by Denecker and his collaborators [Den98, DBM01].

The basic results for the complexity of the well-founded semantics of predicate logic programs can be found in Schlipf [S95] and Fitting [F01]. Complexity results for the stable model semantics of logic programs can be found in [MNR94]. Basically, 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 [S95] and the problem of deciding whether a recursive program has a stable model is Σ^1_1-complete [MNR94].

The main reason for the extremely high complexity of the well-founded semantics for finite predicate logic programs is that Van Gelder's alternating fixed point algorithm to compute the well-founded semantics [V89, V93] must be transfinitely iterated through the recursive ordinals to obtain a fixed point. This type of construction reminded us of a classical construction from topology which has a similar flavor, namely, the problem of finding the Cantor-Bendixson rank of an effectively closed set in 2^ω.

The Cantor-Bendixson derivative first appeared in a paper in 1883 by Bendixson [B1883] in which he proved what is now called the Cantor-Bendixson Theorem based on ideas from Cantor. Rather than state that theorem in its full generality, we shall focus on the space of interest to us which is 2^ω. One puts a topology on 2^ω by defining the basic open sets of the topology to be any set of the form O_τ where τ is a finite sequence of 0s and 1s and O_τ is the set all infinite strings in 2^ω that extend τ. Here a closed set $Q \subseteq 2^\omega$ is effectively closed or is a Π^0_1 class if the complement of Q is a recursively enumerable union of basic open sets in 2^ω. Such a Q can always be thought of as the set of infinite paths through a primitive recursive binary tree. An element $x \in Q$ is said to be **isolated** if there is an open set U such that $Q \cap U = \{x\}$. Q is said to be

perfect if it has no isolated elements. 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.

In this paper, we shall use the recursion theoretic technique of classifying index sets relative to the arithmetic hierarchy to measure complexity. This approach is important since it provides for a finer classification of the complexity of various decision problems. For example, let $\phi_e : \omega \to \omega$ be the partial recursive function computed by the e-th Turing machine and let W_e be the 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. We say I is an *index set* if whenever $\phi_e = \phi_f$, then $e \in I \iff f \in I$. We say that a set $B \subseteq \omega$ is
(i) Σ^0_0 and Π^0_0 if B is recursive,
(ii) Σ^0_n if there is a recursive predicate $R(x, y_1, \ldots, y_n)$ such that

$$x \in B \iff (\exists y_1)(\forall y_2)(\exists y_3)\cdots(Qy_n)R(x, y_1, \ldots, y_n)$$

where Q is $\exists$ if n is odd and $\forall$ if n is even,
(iii) Π^0_n if there is a recursive predicate $R(x, y_1, \ldots, y_n)$ such that

$$x \in B \iff (\forall y_1)(\exists y_2)(\forall y_3)\cdots(Qy_n)R(x, y_1, \ldots, y_n)$$

where Q is $\forall$ if n is odd and $\exists$ if n is even, and
(iv) B is Δ^0_n if it is both Σ^0_n and Π^0_n.
We say that a set A is *Σ^0_n-complete* (*Π^0_n-complete*) if A is Σ^0_n (Π^0_n) and every Σ^0_n (Π^0_n) is many-one reducible to A. Then, for example, it is well known that there is no uniform effective procedure which given e will decide whether W_e is non-empty, finite, or recursive. However, the complexities of deciding whether a given r.e. set W_e is non-empty, finite, or recursive are not the same. That is, consider the index sets $Non = \{e : W_e \text{ is non-empty}\}$, $Fin = \{e : W_e \text{ is finite}\}$, and $Rec = \{e : W_e \text{ is recursive}\}$. 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 [Soa87]. From a practical point of view, if a predicate is Σ^0_n complete or Π^0_n complete for $n > 1$, then we have no way to produce any kind of effective algorithm to determine whether the predicate holds (fails) or even to effectively enumerate all instances for which the predicate holds (fails).

To help us define the index sets of interest to us in this paper, we shall assume that we are given an effective enumeration of all primitive recursive trees $T_0, T_1, \ldots$ and an effective enumeration of all finite predicate logic programs $LP_0, LP_1, \ldots$ over a recursive predicate logic language $\mathcal{L}$ which contains infinitely many constant symbols, infinitely many propositional letters, and for each $n \geq 1$, infinitely many function n-ary function symbols and n-relation symbols. In particular, we shall assume that $\mathcal{L}$ has a constant symbol $\overline{0}$ and a unary function symbol s and we let $\overline{n} = s^n(\overline{0})$ for all $n \in \omega$. For any property $\mathcal{R}$ of finite predicate logic programs, we let $I_{LP}(\mathcal{R}) = \{e : LP_e$ has property $\mathcal{R}\}$. The set $I_{LP}(\mathcal{R})$ is called *index set for property* $\mathcal{R}$ relative to finite predicate logic programs. For any tree $T \subseteq \{0,1\}^*$, we let $[T]$ denote the set of all infinite paths through T. Then it is known that $[T_0], [T_1], \ldots$ is an effective list of all Π^0_1 classes. Then for any property $\mathcal{R}$ of Π^0_1 classes, we let $I_{PC}(\mathcal{R}) = \{e : [T_e]$ has property $\mathcal{R}\}$. The set $I_{PC}(\mathcal{R})$ is called *index set for property* $\mathcal{R}$ relative to Π^0_1 classes.

There has been considerable research on classifying the complexity of index sets of the form $I_{PC}(\mathcal{R})$ for various properties $\mathcal{R}$ concerning the Cantor-Bendixson derivative. One of the main results of this paper will be to show that there is a recursive function f such that for each primitive recursive binary tree T_e, the finite predicate logic program $LP_{f(e)}$ has the property that if λ is either a limit ordinal or zero 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 $LP_{f(e)}$ is equivalent to the complexity of the $\lambda + \alpha$-th derivative of the Π^0_1 class $[T_e]$. 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 $LP_{f(e)}$ will give the well-founded semantics of LP_e at level $\lambda + 2n$. Our correspondence $T_e \to LP_{f(e)}$ will allows us to transfer results about index sets for Π^0_1 classes to produce new complexity results for index sets associated with the well-founded semantics of finite predicate logic programs. For example, we can show that the set of all e such that the true sentences under the well-founded semantics of LP_e is recursive is a Π^1_1-complete set. Thus the problem of deciding whether the well-founded semantics of a finite predicate logic 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 well-founded semantics of LP_e is empty is recursive, the set of e such that the false sentences under the well-founded semantics of LP_e is empty is Π^0_3 complete, and the set of e such the true sentences under the well-founded 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 of finite predicate logic programs 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 [CR98] 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

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

where $p, q_1, \ldots, q_m, r_1, \ldots, r_n$ are atomic formulas in $\mathcal{L}$. Then p is called the head of C and will be denoted by $head(C)$, $\{q_1, \ldots, q_n\}$ is called the positive body of C and will be denoted by $PosBody(C)$, and $\{r_1, \ldots, r_n\}$ is called the negative body of C and will be denoted by $NegBody(C)$. C is called a Horn clause if $NegBody(C) = \emptyset$. A ground atom is an atomic formula without variables and a ground instance of C is a substitution instance of C which has no free variables.

A finite predicate logic program is a finite set of clauses of the form (1). We let $ground(P)$ denote the set of all ground instances of clauses in P. The Herbrand base of P, $H(P)$, is the set of all ground instances of atoms that appear in P. We say that a set of atoms $M \subseteq H(P)$ is a model of a clause $C \in ground(P)$ if either M does not satisfy the body of C or M satisfies the head of C (or both). M is said to be a model of a logic program P if M is a model of each of the clauses of $ground(P)$. P is said to 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 $ground(P)$. That is, given a set I of atoms, we let $T_P(I) = \{p : \exists C = p \leftarrow a_1, \ldots, a_n \in ground(P) : a_1, \ldots, a_n \in I\}$. Then the least model of P, $LM(P)$, equals $T_P \uparrow_\omega (\emptyset) = \bigcup_{n \geq 1} T_P^n(\emptyset)$.

Next assume P is a logic program with negated atoms in the body of some of its clauses. Then following [GL88], we define the *stable models* of P as follows. Assume $M \subseteq H(P)$. The *Gelfond-Lifschitz reduct* of $ground(P)$ by M is a Horn program arising from P by first eliminating those clauses in $ground(P)$ 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 propositional logic 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 finite predicate logic program P. We let $2^{H(p)}$ denote the set of all subsets of $H(P)$ and for any set $M \subseteq H(P)$, let $\overline{M} = H(P) - M$. Then we define the operator $A_P : 2^{H(P)} \rightarrow 2^{H(P)}$ by

$$A_P(M) = LM(GL_M(P)). \tag{2}$$

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 $U_P = A_P^2$ is monotone. Also the operator V_P defined by

$$V_P(M) = \overline{U_P(\overline{M})} \tag{3}$$

is monotone. Next we define U_P^α and V_P^α for any ordinal α by

$$U_p^0(M) = M,\ V_p^0(M) = M,$$
$$U_p^{\alpha+1}(M) = U_P(U_P^\alpha(M)),\ V_p^{\alpha+1}(M) = V_P(V_P^\alpha(M)),$$
$$U_p^\lambda(M) = \bigcup_{\alpha<\lambda} U_P^\alpha(M)),\ \text{and} V_p^\lambda(M) = \bigcup_{\alpha<\lambda} V_P^\alpha(M)) \text{ for } \lambda \text{ a limit ordinal.}$$

It follows from the Knaster-Tarski Theorem [T55] that both U_P and V_P must have least fixed points. Then we can define the set of atoms that are true under the well-founded semantics to be $\mathbb{T}_{wfs}(P) = lpf(U_P)$ and the set of atoms which are false under the well-founded semantics to be $\mathbb{F}_{wfs}(P) = lfp(V_P)$. It is also not difficult to see that $\mathbb{F}_{wfs}(P) = \overline{A_P(\mathbb{T}_{wfs}(P))}$.

Van Gelder [V89, V93] gave the following alternating fixed point algorithm to compute the well-founded semantics which inductively defines sets $\mathbb{F}_\alpha(P)$ and $\mathbb{T}_\alpha(P)$ for all ordinals α. We say that an ordinal α is an *even ordinal* if $\alpha = \lambda + 2n$ where λ is either 0 or a limit ordinal and $n \in \omega$ and α is an *odd ordinal* if $\alpha = \lambda + 2n + 1$ where λ is either 0 or a limit ordinal and $n \in \omega$.

Algorithm
$\mathbb{F}_0(P) := \emptyset$ and $\mathbb{T}_0(P) := A_P(\overline{\mathbb{F}_0}) = LM(GL_{H(P)}(P))$.

$\mathbb{F}_{\alpha+1}(P) = \overline{\mathbb{T}_\alpha}$ and $\mathbb{T}_{\alpha+1}(P) = A_P(\overline{\mathbb{F}_{\alpha+1}(P)}) = LM(GL_{\overline{\mathbb{F}_{\alpha+1}(P)}}(P))$.

For λ a limit ordinal,
$\mathbb{F}_\lambda(P) = \bigcup_{\alpha<\lambda, \alpha \text{ even}} \mathbb{F}_\alpha(P)$ and $\mathbb{T}_\lambda(P) = A_P(\overline{\mathbb{F}_\lambda(P)}) = LM(GL_{\overline{\mathbb{F}_\lambda(P)}}(P))$.

Then $\mathbb{F}_{wfs}(P) = \mathbb{F}_\alpha(P)$ and $\mathbb{T}_{wfs} = \mathbb{T}_\alpha(P)$ where α is the least ordinal such that $\mathbb{F}_\alpha(P) = \mathbb{F}_{\alpha+1}(P)$.

Here is an example of the algorithm which was given by Van Gelder [V93].

Example 1 Let the Herbrand base $H = \{a, b, c, d, e, f, g, h, i\}$ and let the program P be given by the following clauses.

$a \leftarrow c, \neg b$; $\quad b \leftarrow \neg a$; $\quad c$; $\quad d \leftarrow h$;
$d \leftarrow e, \neg f$; $\quad d \leftarrow f, \neg g$; $\quad e \leftarrow d$;
$f \leftarrow e$; $\quad f \leftarrow \neg c$ $\quad i \leftarrow c, \neg d$

Then removing all clauses with negations, $GL_H(P)$ has the clauses
c; $\quad d \leftarrow h$; $\quad e \leftarrow d$; $\quad f \leftarrow e$

Thus $\mathbb{T}_1(P) = \{c\}$. Then $GL_{\mathbb{T}_1(P)}$ has the additional clauses
$a \leftarrow c$; $\quad b$; $\quad d \leftarrow e$; $\quad d \leftarrow f$; $\quad i \leftarrow c$

Thus $\mathbb{T}_2(P) = \{a, b, c, i\}$. Now we lose the first two clauses above, so that $\mathbb{T}_3(P) = \{c, i\}$.

But this means that $GL_{\mathbb{T}_3(P)} = GL_{\mathbb{T}_1(P)}$, so that $\mathbb{T}_4(P) = \mathbb{T}_2(P)$, $\mathbb{T}_5(P) = \mathbb{T}_3(P)$ and so on.

Hence the alternating fixed point has positive facts $\mathbb{T}_{wfs}(P) = \{c, i\}$ and negative facts $\mathbb{F}_{wfs}(P) = \{d, e, f, g, h\}$.

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,

$$\begin{aligned}
\mathbb{F}_{\alpha+2}(P) &= \overline{\mathbb{T}_{\alpha+1}(P)} = \overline{A_P(\overline{\mathbb{F}_{\alpha+1}(P)})} \\
&= \overline{A_P(\mathbb{T}_\alpha(P))} = \overline{A_P(A_P((\overline{\mathbb{F}_\alpha}(P)))} = V_P(\mathbb{F}_\alpha(P)) \text{ and} \\
\mathbb{T}_{\alpha+2}(P) &= A_P(\overline{\mathbb{F}_{\alpha+2}(P)}) = A_P(\mathbb{T}_{\alpha+1}(P)) \\
&= A_P(A_P(\overline{\mathbb{F}_{\alpha+1}(P)})) = A_P(A_P(\mathbb{T}_\alpha(P)) = U_P(\mathbb{T}_\alpha(P)).
\end{aligned}$$

Thus for n finite and λ a limit ordinal, $\mathbb{F}_{2n}(P) = V_P^n(\emptyset)$, $\mathbb{F}_\lambda(P) = V_P^\lambda(\emptyset)$, and $\mathbb{F}_{\lambda+2n}(P) = V_P^{\lambda+n}(\emptyset)$. Similarly, $\mathbb{T}_{2n}(P) = U_P^n(\mathbb{T}_0(P))$, $\mathbb{T}_\lambda(P) = U_P^\lambda(\mathbb{T}_0(P))$, and $\mathbb{T}_{\lambda+2n}(P) = U_P^{\lambda+n}(\mathbb{T}_0(P))$.

Remark: For any finite predicate logic program P, let $Horn(P)$ denote the set of Horn clauses in $ground(P)$. It follows that $GL_{H(P)}(P) = Horn(P)$. Thus $\mathbb{T}_0(P) = LM(Horn(P))$. It is easy to see that $T_0(P)$ is contained in $LM(GL_S(P))$ for any $S \subseteq H(P)$ and hence $T_0(P)$ must be a subset of $U_P(S)$ for any S. Thus the least fixed point of U_P can be found by iterating U_P through the ordinals starting at $T_0(P)$ rather than starting with the empty set.

Then we have the following.

Proposition 1 *Let P be any finite logic program.*

(a) For any even ordinals α and β, if $\alpha < \beta$, then $\mathbb{T}_\alpha(P) \subseteq \mathbb{T}_\beta(P)$ and $\mathbb{F}_\alpha(P) \subseteq \mathbb{F}_\beta(P)$.

(b) For any odd ordinals α and β, if $\alpha < \beta$, then $\mathbb{T}_\beta(P) \subseteq \mathbb{T}_\alpha(P)$ and $\mathbb{F}_\beta(P) \subseteq \mathbb{F}_\alpha(P)$.

(c) For any even ordinal α and any odd ordinal β, $\mathbb{T}_\alpha(P) \subseteq \mathbb{T}_\beta(P)$ and $\mathbb{F}_\alpha(P) \subseteq \mathbb{F}_\beta(P)$.

(d) For any stable model M of P, any even ordinal α and any odd ordinal β, $\mathbb{T}_\alpha(P) \subseteq M \subseteq \mathbb{T}_\beta(M)$ and $\mathbb{F}_\alpha(P) \subseteq \overline{M} \subseteq \mathbb{F}_\beta(M)$.

Proof Part (a) follows from the monotonicity of the operators U_P and V_P and the fact that $\mathbb{F}_0 = \emptyset$. Part (b) follows from part (a) since A_P is anti-monotone.

For part(c), note that $\mathbb{F}_0(P) = \emptyset \subseteq \mathbb{F}_\beta(P)$ for any odd β. Moreover, by our remark preceding the Proposition, $\mathbb{T}_0(P) \subseteq \mathbb{T}_\alpha(P)$ for all α so that $\mathbb{F}_{\alpha+1}(P) = \overline{\mathbb{T}_\alpha(P)} \subseteq \overline{\mathbb{T}_0(P)} = \mathbb{F}_1(P)$ for all α. Since A_P is antimontone, $A_P(\mathbb{T}_\alpha(P)) = \mathbb{T}_{\alpha+1}(P) \subseteq A_P(\mathbb{T}_0(P)) = \mathbb{T}_1(P)$. Thus for all even ordinals β which are not limit ordinals, $\mathbb{T}_\beta(P) \subseteq T_1(P)$. Now suppose that λ is a limit ordinal and for all even $\beta < \lambda$, $\mathbb{T}_\beta(P) \subseteq \mathbb{T}_1(P)$. Then $\mathbb{T}_\lambda(P) = \bigcup_{\beta<\lambda,\beta \text{ even}} \mathbb{T}_\beta(P) \subseteq \mathbb{T}_1(P)$. Thus we can establish by induction that for all even β, $\mathbb{T}_\beta(P) \subseteq \mathbb{T}_1(P)$.

We now proceed by induction. That is, assume that for α even, $\mathbb{F}_\alpha(P) \subseteq \mathbb{F}_\beta(P)$ and $\mathbb{T}_\alpha(P) \subseteq \mathbb{T}_\beta(P)$ for all odd β. Then since U_P and V_P are monotone,

$$\mathbb{F}_{\alpha+2}(P) = V_P(\mathbb{F}_\alpha(P)) \subseteq V_P(\mathbb{F}_\beta(P)) = \mathbb{F}_{\beta+2}(P)$$

and

$$\mathbb{T}_{\alpha+2}(P) = U_P(\mathbb{T}_\alpha(P)) \subseteq U_P(\mathbb{T}_\beta(P)) = \mathbb{T}_{\beta+2}(P)$$

for all odd β. But since $\mathbb{F}_{\alpha+2}(P) \subseteq \mathbb{F}_1(P)$ and $\mathbb{T}_{\alpha+2}(P) \subseteq \mathbb{T}_1(P)$, we have that $\mathbb{F}_{\alpha+2}(P) \subseteq \mathbb{F}_\beta(P)$ and $\mathbb{T}_{\alpha+2}(P) \subseteq \mathbb{T}_\beta(P)$ for all odd β. Now suppose λ is a limit ordinal and for all even ordinals α which are less than λ, $\mathbb{F}_\alpha(P) \subseteq \mathbb{F}_\beta(P)$ and $\mathbb{T}_\alpha(P) \subseteq \mathbb{T}_\beta(P)$ for all odd β. Then clearly, $\mathbb{F}_\lambda(P) = \bigcup_{\alpha<\lambda,\alpha \text{ even}} \mathbb{F}_\alpha(P)$ is a subset of $\mathbb{F}_\beta(P)$ for all odd β and $\mathbb{T}_\lambda(P) = \bigcup_{\alpha<\lambda,\alpha \text{ even}} \mathbb{T}_\alpha(P)$ is a subset of $\mathbb{T}_\beta(P)$ for all odd β.

For part (d), let M be a stable model of P, that is, $M = A_P(M)$. Now by our remark preceding the proposition, $\mathbb{T}_0(P) \subseteq M$. Since A_P is antimontone, we have that $M = A_P(M) \subseteq A_P(\mathbb{T}_0(P)) = \mathbb{T}_1(P)$. Thus we have that $\mathbb{T}_0(P) \subseteq M \subseteq \mathbb{T}_1(P)$. Similarly, we have $\mathbb{F}_0(P) = \emptyset \subseteq \overline{M} \subseteq \overline{\mathbb{T}_0(P)} = \mathbb{F}_1(P)$. We now proceed by induction. That is, suppose α is even and $\mathbb{T}_\alpha(P) \subseteq M \subseteq \mathbb{T}_{\alpha+1}(P)$ and $\mathbb{F}_\alpha(P) \subseteq \overline{M} \subseteq \mathbb{F}_{\alpha+1}(P)$. Then since U_P is monotone and $U_P(M) = M$, we have that $\mathbb{T}_{\alpha+2}(P) \subseteq M \subseteq \mathbb{T}_{\alpha+3}(P)$. Similarly, since V_P is monotone and $V_P(\overline{M}) = \overline{M}$, then $\mathbb{F}_{\alpha+2}(P) \subseteq \overline{M} \subseteq \mathbb{F}_{\alpha+3}(P)$. Next suppose that λ is a limit ordinal and $\mathbb{F}_\alpha(P) \subseteq \overline{M}$ and $\mathbb{T}_\alpha(P) \subseteq M$ for all even ordinals α which are less than λ. Then $\mathbb{F}_\lambda(P) = \bigcup_{\alpha<\lambda,\alpha \text{ even}} \mathbb{F}_\alpha(P)$ is contained in $\overline{M}$ and $\mathbb{T}_\lambda(P) = \bigcup_{\alpha<\lambda,\alpha \text{ even}} \mathbb{T}_\alpha(P)$ is contained in M.

But then $\overline{M} \subseteq \overline{\mathbb{T}_\lambda(P)} = \mathbb{F}_{\lambda+1}(P)$ and $M = A_P(M) \subseteq A_P(\mathbb{T}_\lambda(P)) = \mathbb{T}_{\lambda+1}(P)$. Thus $\mathbb{F}_\lambda(P) \subseteq \overline{M} \subseteq \mathbb{F}_{\lambda+1}(P)$ and $\mathbb{T}_\lambda(P) \subseteq M \subseteq \mathbb{T}_{\lambda+1}(P)$.

With this in mind, we let

$$\widehat{\mathbb{T}}_{wfs}(P) = \bigcap_{odd\alpha} \mathbb{T}_\alpha(P)$$

and

$$\widehat{\mathbb{F}}_{wfs}(P) = \bigcap_{odd\alpha} \mathbb{F}_\alpha(P).$$

It follows that, for any stable model M of P,

$$\mathbb{T}_{wfs}(P) \subseteq M \subseteq \widehat{\mathbb{T}}_{wfs}(P)$$

and

$$\mathbb{F}_{wfs}(P) \subseteq \overline{M} \subseteq \widehat{\mathbb{F}}_{wfs}(P).$$

2.2 Basic Definitions from Recursion Theory

Let $\omega = \{0, 1, 2, \ldots\}$ denote the set of natural numbers, let ω^* denote the set of all finite sequences from ω and let $\{0, 1\}^*$ denote the set of all finite sequences of 0s and 1s. Strings may be coded by natural numbers in the usual fashion. Let $[x, y]$ denote the standard pairing function $\frac{1}{2}(x^2 + 2xy + y^2 + 3x + y)$ and in general $[x_0, \ldots, x_n] = [[x_0, \ldots, x_{n-1}], x_n]$ for all $n \geq 2$. Then a string σ of length n may be coded by $c(\sigma) = [n, [\sigma(0), \sigma(1), \ldots, \sigma(n-1)]]$ and we define the code of the empty sequence $\emptyset$ to be 0. We define the canonical index of any finite set $X = \{x_1 < \cdots < x_n\} \subseteq \omega$ by $can(X) = 2^{x_1} + 2^{x_2} + \cdots + 2^{x_n}$. We define $can(\emptyset) = 0$.

Since we are considering finite programs over our fixed recursive language $\mathcal{L}$, we can use standard Gödel number techniques to assign code numbers to atomic formulas and clauses. That is, we can effectively assign a number to each symbol in $\mathcal{L}$. Then we can think of formulas of $\mathcal{L}$ as sequences of natural numbers so that the code of a formula is just the code of the sequence of numbers associated with the symbols in the formula. Then a clause C as in (1) can be assigned the code of the triple (x, y, z) where x is the code of the conclusion of C, y is the canonical index of the set of codes of $PosBody(C)$, and z is the canonical index of the sets of codes of $NegBody(C)$. It is then not difficult to verify that for any give finite predicate logic program P, the question of whether a given n is the code of a ground atom or a ground instance of a clause in P is a primitive recursive predicate. The key observation to make is that since P is finite and the usual unification algorithm is effective, we can explicitly test whether a given number m is the code of a ground atom or a ground instance of a clause in P without doing any unbounded searches. We say that a set X of ground atoms is recursive, r.e., etc., if the corresponding set of codes of elements of X is recursive, r.e., etc..

Given $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_k)$ in ω^*, 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 $I[\sigma] = \{x \in 2^\omega : \sigma \sqsubseteq x\}$. 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 $S \subseteq \omega^*$ 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 $\{0, 1\}^*$ such that T is closed under initial segments. A tree T is said to be *recursively bounded* if there is a recursive function g such that, for all $\sigma \in T$ and all $i \in \omega$, if $\sigma \in T$, then $\sigma(i) \leq g(i)$. 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 well known that if A is a Π^0_1-class, then $A = [T]$ for some primitive recursive tree $T \subseteq \omega^*$.

To define the index sets of interest to us in this paper, we shall also use Σ^0_α and Π^0_α sets for recursive ordinals α and Σ^1_1 or Π^1_1 sets. These are defined as follows. (See Hinman [Hin78], p. 163ff for details.) The set H of indices of hyperarithmetic sets is first defined. Here the indices of recursive functions all have the form $[n, a]$ where $n \leq 6$; we may assume that ϕ_b is the empty function for any other b. Thus in the definition of the hyperarithmetic sets, we reserve the indices of the form $[7, a]$ to code the r.e. sets.

Definition 1 H is the smallest subset of ω such that, for all a,

(i) $[7, a] \in H$;
(ii) if $\phi_a(n) \in H$ for all n, then $a \in H$.

This is an inductive definition and thus $H = \bigcup_\alpha H^\alpha$ where α ranges over the recursive ordinals. H is a Π^1_1 set and each $a \in H$ is assigned a hyperarithmetic set by the following. Recall that for any $e \in \omega$, ϕ_e is the e'th partial recursive function mapping $\omega \to \omega$ and W_e is the domain of ϕ_e and is the e'th recursively enumerable. This is extended to the hyperarithmetic sets as follows. If $a \notin H$, let $H_a = \emptyset$.

Definition 2 Let $a \in H$. Then

(i) If $a = \langle 7, b \rangle$, then $H_a = W_b$
(ii) If ϕ_a is total, then $H_a = \bigcup_n \omega \setminus H_{\phi_a(n)}$.

The hyperarithmetical hierarchy is defined as follows.

Definition 3 For all ordinals α and all $A \subseteq \omega$,

(i) A is Σ^0_α if $A = H_a$ for some $a \in H^\alpha$;
(ii) A is Π^0_α if $\omega \setminus A$ is Σ^0_α;
(iii) $\Delta^0_\alpha = \Sigma^0_\alpha \cap \Pi^0_\alpha$.

It follows that for limit ordinals λ, A is Σ^0_λ if and only if A is Σ^0_α for some $\alpha < \lambda$.

A set A is said to be Σ^1_1 if there is an arithmetic relation B, i.e. B is either Σ^0_n or Π^0_n for some n, such that, for all x, $x \in A \iff (\exists f \in \omega^\omega)(\forall n) B(x, f \upharpoonright n)$ where $f \upharpoonright n$ is the code of the n-tuple $(f(0), f(1), \ldots, f(n-1))$. A set A is Π^1_1 if its complement is Σ^1_1. 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, $n \in B \iff \phi(n) \in A$. Π^0_α complete, Σ^1_1 complete, and Π^1_1 complete sets are defined similarly. A subset A of ω is said to be D^m_n if it is the difference of two Σ^m_n sets and A is said to be D^m_n *complete* if A is D^m_n and for any D^m_n set B, there is a computable function φ such that, for any n, $n \in B \iff \phi(n) \in A$.

Since finite strings σ may be coded by natural numbers $c(\sigma)$, this also gives us definitions for Σ^0_α sets of strings and for trees, and similarly for the other notions of definability.

To establish our connection between the well-founded semantics and the Cantor-Bendixson derivative, we consider index sets for recursively bounded strong $\Pi^0_{\alpha+1}$ binary classes and also index sets for the cardinality of the Cantor-Bendixson derivatives. For any recursive ordinal α, a recursively bounded strong $\Pi^0_{\alpha+1}$ class is a set of infinite paths through a Σ^0_α binary tree. These problems were first studied in the context of Polish spaces by Kuratowski, see

[Kur70], 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 [C82] 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 [L87].

Here is an example of a non-trivial effectively closed set of rank one.

Example 2 Let B be any infinite subset of ω and let $Q=\{0^\omega\}\cup\{0^n1^\omega : n\in B\}$. This is a closed set and $D(Q)=\{0^\omega\}$. If $B=\omega\setminus A$, where A is a recursively enumerable set, then Q will be a Π^0_1 class. To see this, let A_s be the elements enumerated into A by stage s and define the computable tree $T=\{0^n : n\in\omega\}\cup\{0^n1^s n\notin A_s\}$. We observe that, for each n, $Q\cap I[0^n1]\neq\emptyset$ if and only if $n\notin A$. It follows that Q is a decidable Π^0_1 class if and only if A is recursive.

Example 3 One can modify the example above by letting $Q_1=\{0^n : n\in\omega\}\cup\{0^n1^\omega : n\in\omega\}\cup\{0^n1^{k+1}0^\omega : n\in B\}$, so that $D(Q_1)=Q$. Then we can say that $0^n1^\omega\in D(Q_1)$ if and only if $n\in B$, so that this problem is Σ^0_1 but not recursive.

More complicated examples may be found in [CR99, CRta] to show that in general for a Π^0_1 class Q, $D(Q)$ is a Π^0_3 class and need not be Δ^0_3. In general the set of isolated points will be Σ^0_3. That is, x is isolated in a Π^0_1 class $Q=[T]$ if and only if **there exists** n such that x is the only element of $Q\cap I[x\restriction n]$, which is to say that **for any** extension σ of $x\restriction n$ other than $x\restriction|\sigma|$, σ has only finitely many extensions in the tree T, which is to say that **there exists** m such that σ has no extensions of length m.

Recall that $T_0,T_1,\ldots$ is an effective list of all primitive recursive trees contained in $\{0,1\}^*$ so that $[T_0],[T_1],\ldots$ is an effective list of all Π^0_1 classes.

We can relativize the notions of Σ^0_α and Π^0_α sets and our enumeration of trees for any oracle X. For example, we let π^X_e be the e-th function primitive recursive relative to the oracle X and $T^X_e=\{\emptyset\}\cup\{\sigma : (\forall\tau\prec\sigma)(\pi^X_e(\langle\tau\rangle)=1)\}$. Then for any fixed set X, we let $[T^X_0],[T^X_1],\ldots$ enumerate the binary classes which are Π^0_1 in X. For any property $\mathcal{R}$, let $I^X_P(\mathcal{R})=\{e : [T^X_e]\text{ has property }\mathcal{R}\}$. Similarly if a set is Σ^0_α relative to the oracle X, we shall say that it is a $\Sigma^{0,X}_\alpha$ set. The following result was proved by Cenzer and Remmel [CR98].

Theorem 1 *For any set X,*

1. $\{e : [T^X_e]$ *is empty*$\}$ *is* $\Sigma^{0,X}_1$ *complete,*
2. $\{e : [T^X_e]$ *has cardinality 1*$\}$ *is* $\Pi^{0,X}_2$ *complete.*
3. *For any integer* $c>0$, $\{e : [T^X_e]$ *has cardinality* $>c\}$ *is* $\Sigma^{0,X}_2$ *complete and* $\{e : [T^X_e]$ *has cardinality* $c+1\}$ *is* $D^{0,X}_2$ *complete.*

4. $\{e : [T_e^X] \text{ finite}\}$ *is* $\Sigma_3^{0,X}$ *complete.* □

To classify index sets connected with the transfinite Cantor-Bendixson derivatives of Π_1^0 classes, Cenzer and Remmel [CR98] established a correspondence between the $\Pi_{2\alpha+1}^0$ classes and the α-th Cantor-Bendixson derivatives of Π_1^0 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 $2\lambda - 1 = \lambda$. Note that for limit ordinals λ, we follow the convention that a set is Σ_λ^0 if and only if it is Σ_α^0 for some $\alpha < \lambda$ and is $\Sigma_{\alpha+1}^0$ if it is an effective union of sets which are all Σ_α^0.

Cenzer and Remmel [CR98] proved the following.

Theorem 2 *For any computable ordinal* α

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

Theorem 3 *The following index sets are all* Π_1^1 *complete:*

1. $\{e : K[T_e] \text{ is countable}\} = \{e : K([T_e]) \text{ is empty}\}$.
2. $\{e : K([T_e]) \text{ is } \Delta_1^1\} = \{e : K([T_e]) \text{ is } \Pi_1^1\}$.
3. $\{e : K([T_e]) \text{ is recursive}\}$.

Theorem 4 *There is a* Π_1^0 *class* Q *such that*

1. $rk(Q) = \omega_1^{C\text{-}K}$
2. $\{\sigma : I[\sigma] \cap K(Q) = \emptyset\}$ *is* Π_1^1 *complete.*

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

In this section, we shall define a simple finite predicate logic program P_e for each primitive recursive tree T_e such that for all $n \geq 0$ and λ which is either a recursive limit ordinal or 0,

$$\mathbb{T}_{\lambda+2n}(P_e) = \{\sigma \in \{0,1\}^* : I[\sigma] \cap D^{\lambda+n}([T_e]) = \emptyset\}.$$

This shows that there is a simple connection between the construction of a perfect kernel of Π_1^0 classes and Van Gelder's alternating fixed point construction of the well-founded semantics of finite predicate logic 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.

We shall define a function $f : \omega \to \omega$ by uniformly constructing a finite predicate logic program $P_e = LP_{f(e)}$ depending on T_e. The underlying language of P_e will contain constant symbols $\emptyset$ and $\overline{\emptyset}$ and function symbols L, R, $\overline{L}$ and $\overline{R}$. Here we think of $\emptyset$ as the empty sequence and L and R are two successor functions which may be interpreted as adding 0 or 1 to the end of a sequence. Thus the ground term involving $\emptyset$, L and R can be identified the set of $\sigma \in \{0,1\}^*$. We think of $\overline{\emptyset}$, $\overline{R}$, and $\overline{L}$ as giving us as second copy of $\{0,1\}^*$ so that we shall identify those terms with the set of $\overline{\sigma}$ such $\sigma \in \{0,1\}^*$. In addition, we shall use unary relation symbols seq and $\overline{seq}$, where $seq(x)$ indicates that x is a sequence built up from $\emptyset$ by some applications of L and R, that is, x represents a member of $\{0,1\}^*$. Similarly $\overline{seq}(x)$ indicates that x is a term in the language generated by $\overline{\emptyset}, \overline{L}, \overline{R}$. We shall also have a binary relation $Bar(x,y)$ which is intended to hold if and only x is a term representing some $\sigma \in \{0,1\}^*$ and y is the term representing some $\overline{\sigma}$. This is accomplished by including the following clauses in P_e.

$(A)\ seq(\emptyset) \leftarrow$	$(B)\ \overline{seq}(\overline{\emptyset}) \leftarrow$
$(C)\ seq(L(x)) \leftarrow seq(x)$	$(D)\ seq(R(x)) \leftarrow seq(x)$
$(E)\ \overline{seq}(\overline{L}(x)) \leftarrow \overline{seq}(x)$	$(F)\overline{seq}(\overline{R}(x)) \leftarrow \overline{seq}(x)$
$(G)\ Bar(\emptyset, \overline{\emptyset})$	
$(H)\ Bar(L(x), \overline{L}(y)) \leftarrow Bar(x,y)$	$(I)\ Bar(R(x), \overline{R}(y)) \leftarrow Bar(x,y)$

We shall also need a ternary relation $Con(x,y,z)$ which indicates that z represents the concatenation of x with y. This is only needed for elements of seq and is defined by the following clauses as follows.

$(J)\ Con(x, \emptyset, x)$	
$(K)\ Con(x, L(y), L(z)) \leftarrow Con(x,y,z)$	$(L)\ Con(x, R(y), R(z)) \leftarrow Con(x,y,z)$

A classical result, first explicit in [Sm68] and [AN78] but known a long time earlier in equational form, is that every r.e. relation can be computed by a suitably chosen predicate over the least model of a finite Horn program. Thus we let P_e^- be a finite predicate Horn program such that the least fixed point of P_e consists of the set of $NT(x)$ such that $seq(x)$ and $x \notin T_e$. Finally, we introduce a new predicate In which is designed to capture the perfect kernel of T_e and define the finite predicate logic program $P_e = LP_{f(e)}$ to consist of P_e^- plus clauses (A)-(L) plus the following set of clauses.

(1) $In(x) \leftarrow NT(x)$

(2) $In(x) \leftarrow seq(x), In(L(x)), In(R(x))$

(3) $In(w) \leftarrow \overline{seq}(w), Bar(x,w), seq(x), seq(y), Con(x,y,z), \neg In(L(z)),$
$\neg In(R(z))$

(4) $In(x) \leftarrow seq(x), \overline{seq}(y), Bar(x,y), \neg In(y)$

Let $U = H(P_e)$ denote the Herbrand base of P_e. The intended stable model M of P_e consists of

$$In = \{In(\sigma) : I(\sigma) \cap K([T_e]) = \emptyset\} \cup \{In(\overline{\sigma}) : I[\sigma] \cap K([T_e]) \neq \emptyset\}$$

together with the predicates seq, $\overline{seq}$, Bar, Con, and NT as defined above. Note that these latter predicates are all defined by a Horn program. It follows that for any $S \subseteq U$, $GL_S(P_e)$ always contains all the Horn clauses defining the predicates $seq(x), \overline{seq}(x), Bar(x,y), Con(x,y,z)$ and $NT(x)$. Thus these predicates will always behave as expected in $LM(GL_S(P_e))$. Thus the key clauses are the ones that involve the predicate In which can always be reduced to the following set of clauses when computing $LM(GL_M(P_e))$.

(a) $In(\sigma) \leftarrow$ for $\sigma \notin T_e$,
(b) $In(\sigma) \leftarrow In(\sigma^\frown 0), In(\sigma^\frown 1)$ for all $\sigma, \tau \in \{0,1\}^*$,
(c) $In(\overline{\sigma}) \leftarrow$ for all $\sigma \in \{0,1\}^*$ such that there exists a $\tau \in \{0,1\}^*$ such that $In(\sigma^\frown\tau^\frown 0)$ and $In(\sigma^\frown\tau^\frown 1)$ are both not in M, and
(d) $In(\sigma) \leftarrow$ for all σ such that $In(\overline{\sigma}) \notin M$.

If $In(\sigma) \in M$, then by our definition of M, $In(\overline{\sigma}) \notin M$ so that $In(\sigma) \in LM(GL_M(P_e))$ by rule (d). If $In(\overline{\sigma}) \in M$, 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 $In(\sigma^\frown\tau^\frown 0)$ and $In(\sigma^\frown\tau^\frown 1)$ are not in M, so that $\overline{\sigma} \in LM(GL_M(P_e))$ by clause (c). Thus $M \subseteq LM(GL_M(P_e))$.

On the other hand, if $In(\sigma) \in LM(GL_M(P_e))$, then we can argue by induction on the length of the derivation of $In(\sigma)$ from the one-step provability operator associated with $GL_M(P_e)$ that $In(\sigma) \in M$. That is, if $In(\sigma)$ is derived via a clause of type (a), then $\sigma \notin T_e$, so certainly $In(\sigma) \in M$. If $In(\sigma)$ is derived by a clause (b), then by induction both $In(\sigma^\frown 0)$ and $In(\sigma^\frown 1)$ 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 $In(\sigma) \in M$. If $In(\sigma)$ comes in by clause (d), then $In(\overline{\sigma}) \notin M$, so that $In(\sigma) \in M$. Finally, if $In(\overline{\sigma}) \in LM(GL_M(P_e))$, then, for some $\tau \in \{0,1\}^*$, $In(\sigma^\frown\tau^\frown 0), In(\sigma^\frown\tau^\frown 1) \notin M$. But then $I[\sigma] \cap K([T_e]) \supseteq I[\sigma^\frown\tau^\frown 0] \cap K([T_e]) \neq \emptyset$ so that $In(\overline{\sigma}) \in M$. Thus $LM(GL_M(P_e)) \subseteq M$ and hence M is a stable model.

For the program P given by the tree from Example 2, we see that $\mathbb{T}_0(P) \cap In$ contains all σ which are not in $\{0^n : n \in \omega\} \cup \{0^n 1^k : n \in B\}$ and does not contain any $\overline{\sigma}$. That is, $\mathbb{T}_0(P) \cap In = \{\sigma : I[\sigma] \cap P \neq \emptyset\}$.

It follows from clause (d) above that $\mathbb{T}_1 = GL_M(P)$, where $M = \mathbb{T}_0(P)$, will contain $In(\sigma)$ for all σ together with $In(\overline{\sigma})$ for all σ of the form 0^n. The latter is true since for each n, there is some m such that both $0^{n+m}1$ and $0^{n+m}0$ are in our tree T.

Since $D(Q) = \{0^\omega\}$, the strings of the form 0^n are exactly those which more than one extension in $D(Q)$. Thus $\mathbb{F}_2(P) \cap In = \{\overline{\sigma} : card(I[\sigma] \cap P) \leq 1\}$.

Computing $GL_M(P)$ for $M = \mathbb{T}_1(P)$, we see that $\mathbb{T}_2(P)$ contains $In(\sigma)$ for all σ not of the form 0^n (by clause (d)) and contains no $In(\overline{\sigma})$.

It then follows that $\mathbb{T}_3(P) = \{In(\sigma) : \sigma \in \{0,1\}^*\}$. Every $In(\sigma)$ is in $\mathbb{T}_3(P)$ since no $In(\overline{\sigma})$ is in $\mathbb{T}_2(P)$ and no $In(\overline{\sigma})$ is in $\mathbb{T}_3(P)$ since $\mathbb{T}_2(P)$ contains $In(\sigma ^\frown \tau ^\frown 1)$ for every σ and τ.

It is then easy to see that $\mathbb{T}_4(P) = \mathbb{T}_3(P)$ and this is the fixed point $\mathbb{T}_{wfs}(P)$ of the alternating semantics. Since $D^2(P) = \emptyset$, we have $In \cap \mathbb{T}_4(P) = \{In(\sigma) : I[\sigma] \cap D^2(P) = \emptyset\}$.

The main result of this paper is the following.

Theorem 5 *For all e, all finite n, and λ either a recursive limit ordinal or 0,*

$$In \cap \mathbb{T}_{\lambda+2n}(P_e) = \{In(\sigma) : I[\sigma] \cap D^{\lambda+n}([T_e]) = \emptyset\}, \tag{4}$$

$$In \cap \mathbb{F}_{\lambda+2n+2}(P_e) = \{In(\overline{\sigma}) : card(I[\sigma] \cap D^{\lambda+n}([T_e])) \leq 1\} \text{ and} \tag{5}$$

$$In \cap \mathbb{F}_{\lambda}(P_e) = \{In(\overline{\sigma}) : I[\sigma] \cap D^{\lambda}([T_e]) = \emptyset\} \text{ if } \lambda > 0. \tag{6}$$

Proof We observed above that for any $S \subseteq U = H(P_e)$, $GL_S(P_e)$ always contains all the Horn clauses defining the predicates $seq(x), \overline{seq}(x), Bar(x,y), Con(x,y,z)$ and $NT(x)$. Thus these predicates will always behave as expected in $LM(GL_S(P_e))$. It follows that in computing $LM(GL_S(P_e))$, the clauses (1)-(4) concerning the predicate $In(\cdot)$ are equivalent to the following clauses:

(i) $In(\sigma) \leftarrow$ for $\sigma \notin T_e$,
(ii) $In(\sigma) \leftarrow In(\sigma ^\frown 0), In(\sigma ^\frown 1)$ for all $\sigma, \tau \in \{0,1\}^*$,
(iii) $In(\overline{\sigma}) \leftarrow \neg In(\sigma ^\frown \tau ^\frown 0), \neg In(\sigma ^\frown \tau ^\frown 1)$ for all $\sigma, \tau \in \{0,1\}*$, and
(iv) $In(\sigma) \leftarrow \neg In(\overline{\sigma})$ for all $\sigma \in \{0,1\}^*$.

Fix e and consider the levels of $\mathbb{F}_\alpha(P_e)$ and $\mathbb{T}_\alpha(P_e)$. Note that among the clauses $(i) - (iv)$, $GL_U(P_e)$ has only the Horn clauses (i) and (ii). Now if $I[\sigma] \cap [T_e] = \emptyset$, then by König'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). It is easy to see that if $I[\sigma] \cap [T_e] \neq \emptyset$, then one can not use clauses (i) and (ii) to derive $In(\sigma)$. Thus for $\mathbb{T}_0(P_e) = LM(GL_U(P_e))$, we have

$$In \cap \mathbb{T}_0(P_e) = \{In(\sigma) : I[\sigma] \cap [T_e] = \emptyset\}$$

which establishes the base case for (4).

Next consider $\mathbb{T}_1(P_e)$. Among the clauses $(i) - (iv)$, $GL_{\mathbb{T}_0(P_e)}(P_e)$ consists of the Horn clauses (i) and (ii) together with the following two families of clauses. First there are clauses $In(\overline{\sigma}) \leftarrow$ 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 $card(I[\sigma] \cap [T_e]) \geq 2$. Second, there are clauses $In(\sigma) \leftarrow$ for all σ such that $In(\overline{\sigma}) \notin \mathbb{T}_0(P_e)$, which is to say for all $\sigma \in \{0,1\}^*$. Thus

$$\begin{aligned} &In \cap \mathbb{T}_1(P_e) = In \cap LM(GL_{\mathbb{T}_0(P_e)}(P_e)) = \\ &\{In(\sigma) : \sigma \in \{0,1\}^*\} \cup \{In(\overline{\sigma}) : card(I[\sigma] \cap [T_e]) \geq 2\}. \end{aligned}$$

This means that

$$\begin{aligned} In \cap \mathbb{F}_2(P_e) &= \{In(\sigma), In(\overline{\sigma}) : \sigma \in \{0,1\}^*\} - (In \cap LM(GL_{\mathbb{T}_0(P_e)}(P_e))) \\ &= \{In(\overline{\sigma}) : card(I[\sigma] \cap [T_e]) \le 1\}. \end{aligned}$$

This establishes the base case for (5).

Next we observe that (6) follows by induction and compactness. That is, suppose that λ is a limit ordinal. Then since $D_\lambda(Q) = \bigcap_{\alpha<\lambda} D_\alpha(Q)$ and 2^ω is compact, it follows that for any closed set $Q \subseteq 2^\omega$ and any $\sigma \in 2^\omega$, $I[\sigma] \cap (D_\lambda(Q)) = \emptyset$ if and only if there is some $\alpha < \lambda$ such that $card(I[\sigma] \cap D_\alpha(Q)) \le 1$ if and only if there is some ordinal $\beta < \lambda$, which is either a limit ordinal or 0, and some $n \in \omega$, $card(I[\sigma] \cap D_{\beta+2n+2}(Q)) \le 1$. But then

$$\begin{aligned} In \cap \mathbb{F}_\lambda(P_e) &= In \cap \left(\bigcup_{\beta<\lambda, \beta \text{ a limit or } 0, n \ge 0} \mathbb{F}_{\beta+2n}(P_e) \right) \\ &= In \cap \left(\bigcup_{\beta<\lambda, \beta \text{ a limit or } 0, n \ge 0} \mathbb{F}_{\beta+2n+2}(P_e) \right) \\ &= \bigcup_{\beta<\lambda, \beta \text{ a limit or } 0, n \ge 0} (In \cap \mathbb{F}_{\beta+2n+2}(P_e)) \\ &= \bigcup_{\beta<\lambda, \beta \text{ a limit or } 0, n \ge 0} \{In[\overline{\sigma}] : card(I[\sigma] \cap D_{\beta+n}([T_e])) \le 1\} \\ &= \{In[\overline{\sigma}] : I[\sigma] \cap D_\lambda([T_e]) = \emptyset\}. \end{aligned}$$

Here the second equality holds because $\mathbb{F}_{\beta+2n}(P_e) \subseteq \mathbb{F}_{\beta+2n+2}(P_e)$ by part (a) of Proposition 1.

Similarly, we can use induction to prove the special case of (4) when λ is a limit ordinal and $n = 0$. That is,

$$\begin{aligned} In \cap \mathbb{T}_\lambda(P_e) &= In \cap \left(\bigcup_{\beta<\lambda, \beta \text{ a limit or } 0, n \ge 0} \mathbb{T}_{\beta+2n}(P_e) \right) \\ &= \bigcup_{\beta<\lambda, \beta \text{ a limit or } 0, n \ge 0} (In \cap \mathbb{T}_{\beta+2n}(P_e)) \\ &= \bigcup_{\beta<\lambda, \beta \text{ a limit or } 0, n \ge 0} \{In(\sigma) : I[\sigma] \cap D_{\beta+n}([T_e])) = \emptyset\} \\ &= \{In(\sigma) : I[\sigma] \cap D_\lambda([T_e]) = \emptyset\}. \end{aligned}$$

Next assume that λ is a limit ordinal and that for all $\sigma \in \{0,1\}^*$, $In(\sigma) \notin \mathbb{F}_\lambda(P_e)$ and $In(\overline{\sigma}) \in \mathbb{F}_\lambda(P_e) \iff I[\sigma] \cap D^\lambda([T_e]) = \emptyset$. Then for all $\sigma \in \{0,1\}^*$, $In(\sigma) \in \overline{\mathbb{F}_\lambda(P_e)}$ and $In(\overline{\sigma}) \in \overline{\mathbb{F}_\lambda(P_e)} \iff I[\sigma] \cap D^\lambda([T_e]) \neq \emptyset$. Hence among the clauses (i)-(iv), $GL_{\overline{\mathbb{F}_\lambda(P_e)}}(P_e)$ contains the clauses (i) and (ii) plus the set clauses $In(\sigma) \leftarrow$ 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

$$In \cap \mathbb{T}_\lambda(P_e) = In \cap LM(GL_{\overline{\mathbb{F}_\lambda(P_e)}}(P_e)) = \{In(\sigma) : I[\sigma] \cap D^\lambda([T_e]) = \emptyset\}.$$

Then among the clauses (i)-(iv), $GL_{\mathbb{T}_\lambda(P_e)}(P_e)$ contains the clauses (i) and (ii) plus all clauses of the from $In(\sigma) \leftarrow$ for $\sigma \in \{0,1\}^*$ plus all clauses of the form $In(\overline{\sigma}) \leftarrow$ such that there exists a τ such that $In(\sigma^\frown\tau^\frown 0)$ and $IN(\sigma^\frown\tau^\frown 1)$ are not in $\mathbb{T}_\lambda(P_e)$. But if $In(\sigma^\frown\tau^\frown 0)$ and $In(\sigma^\frown\tau^\frown 1)$ are not in $\mathbb{T}_\lambda(P_e)$, then $I[\sigma^\frown\tau^\frown 0] \cap D^\lambda([T_e]) \neq \emptyset$ and $I[\sigma^\frown\tau^\frown 1] \cap D^\lambda([T_e]) \neq \emptyset$ which is equivalent to saying that $card(I[\sigma] \cap D^\lambda([T_e])) \geq 2$. Thus

$$In \cap \mathbb{T}_{\lambda+1}(P_e) = \{In(\sigma) : \sigma \in \{0,1\}^*\} \cup \{In(\overline{\sigma}) : card(I[\sigma] \cap D^\lambda([T_e])) \geq 2\}.$$

Hence $In \cap \mathbb{F}_{\lambda+2}(P_e) = \{In(\overline{\sigma}) : card(I[\sigma] \cap D^\lambda([T_e]) \leq 1\}$.

Finally suppose that for $n \geq 1$,

$$In \cap \mathbb{F}_{\lambda+2n}(P_e) = \{In(\overline{\sigma}) : card(I[\sigma] \cap D^{\lambda+n-1}([T_e])) \leq 1\}.$$

Then among the clauses (i)-(iv), $GL_{\overline{\mathbb{F}_{\lambda+2n}(P_e)}}(P_e)$ consists of the clauses (i) and (ii) for all $\sigma \in \{0,1\}^*$ and the clause $In(\sigma) \leftarrow$ for all $\sigma \in \{0,1\}^*$ such that $I[\sigma] \cap D^{\lambda+n-1}([T_e]) \leq 1\}$. It follows that

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

which equals $\{In(\sigma) : I[\sigma] \cap D^{\lambda+n}([T_e]) \text{ is empty}\}$ as desired. But then among the clauses (i)-(iv), $GL_{\mathbb{T}_{\lambda+2n}(P_e)}(P_e)$ consists of clauses (i) and (ii) for all $\sigma \in \{0,1\}^*$ plus the clauses $In(\sigma) \leftarrow$ for all $\sigma \in \{0,1\}^*$ plus the clauses $In(\overline{\sigma}) \leftarrow$ for all $\sigma \in \{0,1\}^*$ 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 for all $\sigma \in \{0,1\}^*$, $In(\sigma) \in LM(GL_{\mathbb{T}_{\lambda+2n}(P_e)})$ and

$$In(\overline{\sigma}) \in LM(GL_{\mathbb{T}_{\lambda+2n}(P_e)}(P_e)) \iff card(I[\sigma] \cap D^{\lambda+n}([T_e])) \geq 2\}.$$

Since $\mathbb{F}_{\lambda+2n+2} = U - LM(GL_{\mathbb{T}_\alpha(P_e)}(P_e))$, it follows that

$$In \cap \mathbb{F}_{\lambda+2n+2} = \{In(\overline{\sigma}) : card(I[\sigma] \cap D^{\lambda+n}([T_e])) \leq 1\},$$

as desired. This completes the inductive proof of (4) and (5).

We then have the following corollary.

Corollary 1 *For all e,*

$$In \cap T_{wfs}(P_e) = \{In(\sigma) : I[\sigma] \cap K([T_e]) = \emptyset\} \text{ and} \tag{7}$$
$$In \cap F_{wfs}(P_e) = \{In(\overline{\sigma}) : I[\sigma] \cap K([T_e]) = \emptyset\}. \tag{8}$$

Proof First by Theorem 5, we have that

$$\begin{aligned} In \cap T_{wfs}(P_e) &= \bigcup_{\lambda \text{ a limit}, n \geq 0} In \cap \mathbb{T}_{\lambda+2n}(P_e) \\ &= \bigcup_{\lambda \text{ a limit}, n \geq 0} \{In(\sigma) : I[\sigma] \cap D^{\lambda+n}([T_e]) = \emptyset\} \\ &= \{In(\sigma) : I[\sigma] \cap K([T_e]) = \emptyset\}. \end{aligned}$$

Similarly,

$$\begin{aligned} In \cap F_{wfs}(P_e) &= \bigcup_{\lambda \text{ a limit}, n \geq 0} In \cap \mathbb{F}_{\lambda+2n}(P_e) \\ &= \bigcup_{\lambda \text{ a limit}, n \geq 0} In \cap \mathbb{F}_{\lambda+2n+2}(P_e) \\ &= \bigcup_{\lambda \text{ a limit}, n \geq 0} \{In(\overline{\sigma}) : card(I[\sigma] \cap D^{\lambda+n}([T_e]) \leq 1\} \\ &= \{In(\overline{\sigma}) : I[\sigma] \cap K([T_e]) = \emptyset\}. \end{aligned}$$

In the following theorem, we consider the arithmetical complexity of subsets of $\omega \times \omega$. Here we identify each ground atom over the recursive language $\mathcal{L}$ with its code. Thus an $M \subseteq H(P)$ can be thought of as a set of natural numbers.

Theorem 6 *Let $\mathbb{T}_{e,\alpha} = \mathbb{T}_\alpha(P_e)$ and $\mathbb{F}_{e,\alpha} = \mathbb{F}_\alpha(P_e)$ be the sequence of sets defined in the alternating fixpoint algorithm to compute the well-founded semantics for the finite predicate logic program P_e constructed from the primitive recursive tree T_e. Then for any finite n and any λ which is either 0 or a recursive limit ordinal,*

1. $\{\langle e,p\rangle : p \in \mathbb{T}_{e,\lambda+2n}\}$ *is a* $\Sigma^0_{\lambda+2n+1}$ *complete set,*
2. $\{\langle e,p\rangle : p \in \mathbb{F}_{e,\lambda+2n+1}\}$ *is a* $\Pi^0_{\lambda+2n+1}$ *complete set,*
3. $\{\langle e,p\rangle : p \in \mathbb{F}_{e,\lambda+2n+2}\}$ *is* $\Pi^0_{\lambda+2n+2}$ *a complete set, and*
4. $\{\langle e,p\rangle : p \in \mathbb{T}_{e,\lambda+2n+1}\}$ *is* $\Sigma^0_{\lambda+2n+2}$ *a complete set.*

Proof Note that (2) follows from (1) since $\mathbb{F}_{\lambda+2n+1}(P_e) = U - \mathbb{T}_{\lambda+2n}(P_e)$ and (4) follows from (3) since $\mathbb{F}_{\lambda+2n+2}(P_e) = U - \mathbb{T}_{\lambda+2n+1}(P_e)$.

Note that $\mathbb{T}_0(P_e) = LM(Horn(P_e))$ is Σ^0_1. In general, the operator $A_{P_e}(M) = LM(GL_M(P_e))$ is Σ^0_1 in M so that $U_{P_e}(M) = A_{P_e}(A_{P_e}(M))$ is Σ^0_2 in M. Similarly, the operator $V_P(M)$ is Π^0_2 in M.

This allows us to prove that for all $n \geq 0$ $\mathbb{T}_{2n}$ is Σ^0_{2n+1} and $\mathbb{F}_{2n+2}$ is Π^0_{2n+2} and that for all recursive limit ordinals λ, $\mathbb{T}_{\lambda+2n}(P_e)$ is $\Sigma_{\lambda+2n+1}$ and $\mathbb{F}_{\lambda+2n+2}(P_e)$ is $\Pi_{\lambda+2n+2}$ uniformly in e.

That is, if λ is a recursive limit ordinal, then

$$\mathbb{T}_\lambda(P_e) = \bigcup_{\alpha<\lambda, \alpha \text{ even}} \mathbb{T}_\alpha(P_e)$$

and

$$\mathbb{F}_{\lambda}(P_e) = \bigcup_{\alpha<\lambda, \alpha \text{ even}} \mathbb{F}_{\alpha}(P_e)$$

are $\Sigma^0_{\lambda+1}$ by standard inductive definability results in [Hin78]. But then $\mathbb{T}_{\lambda+1}(P_e)$ is $\Sigma^0_{\lambda+2}$ and $\mathbb{F}_{\lambda+2}(P_e)$ is $\Pi^0_{\lambda+2}$. Finally if $\mathbb{T}_{\lambda+2n}(P_e)$ is $\Sigma^0_{\lambda+2n+1}$, then $\mathbb{T}_{\lambda+2n+2}(P_e) = U_{P_e}(\mathbb{T}_{\lambda+2n})$ is $\Sigma^0_{\lambda+2n+3}$. Similarly, if $\mathbb{F}_{\lambda+2n}(P_e)$ is $\Pi^0_{\lambda+2n+1}$, then $\mathbb{F}_{\lambda+2n+2}(P_e) = V_{P_e}(\mathbb{F}_{\lambda+2n})$ is $\Pi^0_{\lambda+2n+3}$.

The completeness results follows from Theorems 2 and 5. We will illustrate the proof for infinite ordinals. By the proof of Theorem 5, there is a recursive function f such that the program P_e corresponding to tree T_e in Theorem 5 is $LP_{f(e)}$. Then for any recursive limit ordinal λ and any finite n,

$$I[\emptyset] \cap D^{\lambda+n}([T_e]) = \emptyset \iff D^{\lambda+n}([T_e]) = \emptyset.$$

By Theorem 2, this is a $\Sigma^0_{\lambda+2n+1}$ complete relation on e. But then by Theorem 5,

$$D^{\lambda+n}([T_e]) = \emptyset \iff In(\emptyset) \in T_{\lambda+2n}(LP_{f(e)}).$$

This reduction demonstrates that $\{(e,p) : p \in \mathbb{T}_{\lambda+2n}(P_e)\}$ is $\Sigma_{\lambda+2n+1}$ complete. Similarly we have

$$card(D^{\lambda+n}([T_e])) \leq 1 \iff In(\overline{\emptyset}) \in \mathbb{F}_{\lambda+2n+2}(LP_{f(e)}),$$

which shows that $\{(e,p) : p \in \mathbb{F}_{\lambda+2n+1}(P_e)\}$ is $\Pi_{\lambda+2n+2}$ complete.

We next apply Theorem 3 and Theorem 5 to derive the following index set results for the well-founded semantics.

Theorem 7 *Let R be any infinite and coinfinite recursive subset of U. Then the following index sets are all Π^1_1 complete:*

(i) $\{e : \mathbb{T}_{wfs}(LP_e)$ *is recursive*$\}$
(ii) $\{e : R \subseteq \mathbb{T}_{wfs}(LP_e)\}$, *and*
(iii) $\{e : \mathbb{T}_{wfs}(LP_e)$ *is* $\Delta^1_1\}$.

Proof The upper bound on the complexity follows from the fact that $\mathbb{T}_{wfs}(LP_e)$ can be obtained from the closure of a Σ^0_2 monotone inductive operator. Therefore $\mathbb{T}_{wfs}(LP_e)$ is Δ^1_1 if and only if there exists a countable α such that the inductive operator closes at stage α and, hence, $\mathbb{T}_{\alpha}(LP_e) = \mathbb{T}_{\alpha+2}(LP_e)$ and $\mathbb{F}_{\alpha}(LP_e) = F_{\alpha+2}(LP_e)$. This is a Π^1_1 condition by the Stage Comparison Theorem [Hin78], p. 105.

It follows from the proof of Theorem 5 that there is a 1:1 recursive function f such that the program P_e corresponding to the primitive recursive tree T_e is $LP_{f(e)}$. Since

$$\mathbb{T}_{wfs}(LP_{f(e)}) = \{In(\sigma) : I[\sigma] \cap K([T_e]) = \emptyset\},$$

it is easy to see that $K([T_e])$ is recursive (Δ^1_1) if and only if $\mathbb{T}_{wfs}(LP_{f(e)})$ is recursive (Δ^1_1). Hence f shows that $\{e : K([T_e])$ is recursive$\}$ is 1:1 reducible to $\{e : \mathbb{T}_{wfs}(LP_e) \text{ is recursive}\}$ and $\{e : K([T_e]) \text{ is } \Delta^1_1\}$ is 1:1 reducible to $\{e : \mathbb{T}_{wfs}(LP_e) \text{ is } \Delta^1_1\}$. Thus the Π^1_1-completeness for parts (i) and (iii) follow from Theorem 3. For the Π^1_1-completeness of part (ii), note that $K([T_e]) = \emptyset$ if and only if $In(\sigma) \in \mathbb{T}_{wfs}(LP_{f(e)})$ for all $\sigma \in \{0,1\}^*$. 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 is the recursive set of codes of all $In(\sigma)$ such that $\sigma \in \{0,1\}^*$. But given, any recursive set R which is infinite and coinfinite, we can construct a coding scheme such that R equals the set of codes of all $In(\sigma)$ such that $\sigma \in \{0,1\}^*$.

By combining Theorem with Theorem 5, we obtain the following result which is essentially due to Schlipf [S95].

Theorem 8 *There is a finite predicate logic program P such that the least ordinal α such that $U^{\alpha}_P(\emptyset) = U^{\alpha+1}_P(\emptyset)$ is $\omega_1^{C\text{-}K}$ and $\mathbb{T}_{wfs}(P)$ is a Π^1_1 complete set.*

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

In this section, we will derive a number of index sets results for finite predicate logic programs whose well-founded semantics is extremely simple. First we consider the problem of classifying the index sets for the properties of having $\mathbb{T}_{wfs}(LP_e)$ and/or $\widehat{\mathbb{T}}_{wfs}(LP_e)$ be empty.

Theorem 9 $\{e : \mathbb{T}_0(LP_e) = \mathbb{T}_1(LP_e) = \emptyset\} = \{e : \widehat{\mathbb{T}}_{wfs}(LP_e) = \emptyset\}$ *is recursive.*

Proof Observe that for any finite predicate logic program P,

$$\mathbb{T}_0(P) \subseteq \mathbb{T}_{wfs}(P) \subseteq \widehat{\mathbb{T}}_{wfs}(P) \subseteq \mathbb{T}_1(P).$$

so that if $\mathbb{T}_1(P) = \emptyset$, then $\mathbb{T}_0(P) = \mathbb{T}_{wfs}(P) = \widehat{\mathbb{T}}_{wfs}(P) = \emptyset$.

First assume that $\mathbb{T}_0(LP_e) = \mathbb{T}_1(LP_e) = \emptyset$. Now $\mathbb{T}_0(LP_e) = LM(GL_{H(LP_e)}(LP_e)) = \emptyset$ if and only if there are no Horn clauses in LP_e, that is, the finite program LP_e has no clauses whose negative body is empty. This is clearly a recursive condition. But if $\mathbb{T}_0(LP_e) = \emptyset$, then $\mathbb{T}_1(LP_e) = LM(GL_{\emptyset}(LP_e))$. Thus $\mathbb{T}_1(LP_e) = \emptyset$ if and only if LP_e has no clauses whose positive body is empty which is also a recursive condition. Thus $\{e : \mathbb{T}_1(LP_e) = \mathbb{T}_1(LP_e) = \emptyset\}$ is recursive. Clearly, if $\mathbb{T}_0(P) = \mathbb{T}_1(P) = \emptyset$, then $\widehat{\mathbb{T}}_{wfs}(P) = \emptyset$.

Next suppose that $\widehat{\mathbb{T}}_{wfs}(P) = \emptyset$. Then we know that $\mathbb{T}_{wfs}(P) = \emptyset$ so that $\mathbb{T}_{\alpha}(P) = \emptyset$ for all even α. We claim that this condition forces $\mathbb{T}_1(P) = \emptyset$. That is, suppose for a contradiction, $\mathbb{T}_1(P) = LM(GL_{\emptyset}(P)) = A \neq \emptyset$. Then $\mathbb{F}_2(P) = \overline{A}$ and $\mathbb{T}_2(P) = LM(GL_A(P)) = \emptyset$. But then $\mathbb{F}_3(P) = H(P)$, and $\mathbb{T}_3(P) = LM(GL_{\emptyset}(P)) = A$. It is then easy to prove by induction that $\mathbb{T}_{\alpha}(P) = \emptyset$ if α is even and $\mathbb{T}_{\alpha}(P) = A$ is α is odd which would imply that $\widehat{\mathbb{T}}_{wfs}(P) = A$ contradicting our assumption that $\widehat{\mathbb{T}}_{wfs}(P) = \emptyset$. Thus we have shown that $\widehat{\mathbb{T}}_{wfs}(P) = \emptyset$ if and only if $\mathbb{T}_0(P) = \mathbb{T}_1(P) = \emptyset$.

Theorem 10 *$\{e : \mathbb{T}_{wfs}(LP_e) = \emptyset\}$ is Π^0_2 complete.*

Proof It is clear that $\mathbb{T}_{wfs}(LP_e) = \emptyset$ if and only if $\mathbb{T}_\alpha(LP_e) = \emptyset$ for all even α. THen we must have that $\mathbb{T}_0(LP_e) = \emptyset$. As above, $\mathbb{T}_0(LP_e) = \emptyset$ if and only if there are no Horn clauses in LP_e which is a recursive condition. But then $GL_\emptyset(LP_e)$ is a recursive program so $\mathbb{T}_1(LP_e) = LM(GL_\emptyset(LP_e))$ is an r.e. set and, therefore, $\mathbb{T}_2(LP_e) = LM(GL_{\mathbb{T}_1(LP_e)}(LP_e))$ is a Σ^0_2 set. It follows that the condition $\mathbb{T}_2(LP_e) = \emptyset$ is now Π^0_2.

For the completeness, we give a reduction to the Π^0_2 complete set $\{e : W_e = \omega\}$. Let H_e be a finite Horn program which contains the constant symbol $\overline{0}$, the unary function symbol s, and a predicate symbol R such that $R(\overline{n}) \in lfp(H_e) \iff n \in W_e$. Let $LP_{g(e)}$ consist of the following clauses, where b is an atom that does not occur in H_e:

(i) $p \leftarrow q_1, \ldots, q_m, \neg b$ for each clause $C = p \leftarrow q_1, \ldots, q_m$ of H_e and

(ii) $b \leftarrow \neg R(x)$.

Note $GL_U(LP_{g(e)})$ is the empty program so that $\mathbb{T}_0(LP_{g(e)}) = \emptyset$. It then follows that $GL_{\mathbb{T}_0(LP_{g(e)})}(LP_{g(e)}) = GL_\emptyset(LP_{g(e)})$ has all clauses of H_e, plus the clause $b \leftarrow$. Thus $\mathbb{T}_1(LP_{g(e)})$ will contain b and it will contain $R(\overline{n})$ if and only $n \in W_e$. There are two cases in the determination of $\mathbb{T}_2(LP_{g(e)})$. If $W_e = \omega$, then $GL_{\mathbb{T}_1(LP_{g(e)})}(LP_{g(e)})$ will be the empty program so that $\mathbb{T}_2(LP_{g(e)}) = \emptyset$. If $W_e \neq \omega$, then for some n_0, $n_0 \notin W_e$ in which case $R(\overline{n_0}) \notin \mathbb{T}_1(LP_{g(e)})$. Thus the clause $b \leftarrow$ is in $GL_{\mathbb{T}_1(LP_{g(e)})}(LP_{g(e)})$. But then $b \in \mathbb{T}_2(LP_{g(e)})$ so that $\mathbb{T}_2(LP_{g(e)}) \neq \emptyset$. It follows that $a \in \{e : W_e = \omega\}$ if and only if $\mathbb{T}_0(LP_{g(a)}) = \mathbb{T}_2(LP_{g(a)}) = \emptyset$ if and only if $\mathbb{T}_{wfs}(LP_{g(a)}) = \emptyset$. Hence the set $\{e : \mathbb{T}_{wfs}(LP_e) = \emptyset\}$ is Π^0_2-complete.

We can ask a similar questions the properties of having $\mathbb{F}_{wfs}(LP_e)$ or $\widehat{\mathbb{F}}_{wfs}(LP_e)$ be empty. Here the results are a bit different.

Theorem 11 *$\{e : \mathbb{F}_1(LP_e) = \emptyset\} = \{e : \widehat{\mathbb{F}}_{wfs}(LP_e) = \emptyset\}$ is Π^0_2 complete.*

Proof For any finite predicate logic program P, $\mathbb{F}_1(P) = \emptyset$ if and only if $\mathbb{T}_0(P) = LM(GL_{H(P)}(P)) = H(P)$, where $H(P)$ is the Herbrand base of P. Note that $GL_{H(P)}(P)$ is a recursive program so that $\mathbb{T}_0(P)$ is r.e. and, hence, the predicate that $\mathbb{T}_0(P) = H(P)$ is Π^0_2. Thus the predicate that $\mathbb{F}_1(LP_e) = \emptyset$ is a Π^0_2 predicate. Now it is easy to see by induction, that if $\mathbb{F}_1(LP_e) = \emptyset$, then $\mathbb{F}_\alpha(LP_e) = \emptyset$ for all α. Thus if $\mathbb{F}_1(LP_e) = \emptyset$, then $\widehat{\mathbb{F}}_{wfs}(LP_e) = \emptyset$.

Now suppose that $\widehat{\mathbb{F}}_{wfs}(LP_e) = \emptyset$. Then we claim that $\mathbb{F}_1(LP_e) = \emptyset$. For a contradiction, suppose that $\mathbb{F}_1(LP_e) = A \neq \emptyset$. Then

$$\mathbb{T}_0(LP_e) = LM(GL_{H(LP_e)}(LP_e)) = H(LP_e) - A.$$

But $\mathbb{F}_\alpha(LP_e) \subseteq \mathbb{F}_{wfs}(LP_e) \subseteq \widehat{\mathbb{F}}_{wfs}(LP_e)$ for all even α so that we must have that $\mathbb{F}_2(LP_e) = \emptyset$. This means that $\mathbb{T}_1(LP_e) = LM(GL_{H(LP_e)-A}(LP_e)) =$

$H(LP_e)$. It is then easy to prove by induction that $\mathbb{T}_\beta(LP_e) = LM(GL_{H(LP_e)-A}(LP_e)) = H(LP_e)$ for all odd β and $\mathbb{T}_\alpha(LP_e) = LM(GL_{H(LP_e)}(LP_e)) = H(LP_e) - A$ for all even α. This implies that $F_\beta(LP_e) = A$ for all odd β and hence $\widehat{\mathbb{F}}_{wfs}(LP_e) = A$. Thus if $\widehat{\mathbb{F}}_{wfs}(LP_e) = \emptyset$, then $\mathbb{F}_1(LP_e) = \emptyset$.

For the completeness, we again give a reduction to the Π^0_2 complete set $\{e : W_e = \omega\}$. Let K_e be a Horn program with constant symbol $\overline{0}$, unary predicate s, and predicate R such that, for all n, $R(\overline{n}) \in lfp(K_e) \iff n \in W_e$. Note that K_e may have other predicates, but one can construct K_e so that only ground terms are $\overline{0}$ and $s^n(\overline{0}) = \overline{n}$.

Let h be the recursive function such that $LP_{h(e)}$ consists of K_e together with the clauses

(a) $Q(x_1, \ldots, x_k) \ \leftarrow \ R(x_1), R(x_2), \ldots, R(x_k)$ for all predicates Q of K_e which are different from R.

Since $LP_{h(e)}$ is a Horn program, it is easy to prove by induction that $\mathbb{T}_\alpha(LP_{h(e)}) = lfp(LP_{h(e)})$ for all α. Now suppose that $W_e = \omega$. Then $R(\overline{n}) \in lfp(LP_{h(e)})$ for every $n \in \omega$. Hence the clauses in (a) will allow us to show that $\mathbb{T}_0(LP_{h(e)}) = lfp(LP_{h(e)}) = H((LP_{h(e)})$ and $\mathbb{F}_1(LP_{h(e)}) = \emptyset$.

Next suppose that $W_e \neq \omega$. Then some n_0, $R(\overline{n_0}) \notin lfp(LP_{h(e)})$ and hence $R(\overline{n_0}) \notin lpf(LP_{h(e)})$ so that $R(\overline{n_0}) \in \mathbb{F}_1(LP_{h(e)})$. Thus $a \in \{e : W_e = \omega\} \iff h(a) \in \{e : \mathbb{F}_1(LP_e) = \emptyset\}$. Hence $\{e : \mathbb{F}_1(LP_e) = \emptyset\}$ is Π^0_2 complete.

Theorem 12 *$\{e : \mathbb{F}_{wfs}(LP_e) = \emptyset\}$ is Π^0_3 complete.*

Proof It is easy to see that $\mathbb{F}_{wfs}(LP_e) = \emptyset$ if and only if $\mathbb{F}_2(LP_e) = \emptyset$. Note that $\mathbb{T}_0(LP_e) = LM(GL_{H(LP_e)}(LP_e))$ is r.e. so that $\mathbb{T}_1(LP_e) = LM(GL_{\mathbb{T}_0(LP_e)}(LP_e))$ is Σ^0_2. Thus $\mathbb{F}_2(LP_e)$ is a Π^0_2 set. It follows that the predicate $\mathbb{F}_2(LP_e) = \emptyset$ is Π^0_3.

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

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

Let $R_e(m, n, p)$ be the predicate $R(e, m, n, p)$ and let H_e be a Horn program with predicate $R_e(\cdot, \cdot, \cdot)$ such that the least model of H_e defines the predicate R_e. That is, H_e has a constant term $\overline{0}$ and a unary function symbol s, and ternary predicate R_e such that in the least model of H_e, $R_e(\overline{m}, \overline{n}, \overline{p})$ holds if and only if $R_e(m, n, p)$ holds. Note that H_e may have other predicates, but one can construct H_e so that only ground terms are $\overline{0}$ and $s^n(\overline{0}) = \overline{n}$. Define the program $T_e = LP_{q(e)}$ to consist of H_e together with the following rules where A, and B are new predicates:

(i) $B(x, y) \ \leftarrow \ R_e(x, y, z)$
(ii) $A(x) \ \leftarrow \ \neg B(x, y)$
(iii) $Q(x_1, \ldots, x_k) \ \leftarrow \ A(x_1), A(x_2), \ldots, A(x_k)$ for all predicates Q of H_e.
(iv) $B(x, y) \ \leftarrow \ A(x), A(y)$.

Then it is easy to see that, for any m and n,

$$B(\overline{m}, \overline{n}) \in \mathbb{T}_0(LP_{q(e)}) \iff (\exists p) R_e(m, n, p).$$

It follows that $GL_{\mathbb{T}_0(LP_{q(e)})}(LP_{q(e)})$ will have rules (i), (iii), and (iv) together with rules $A(\overline{m}) \leftarrow$ for all m such that $(\exists n)(\forall p)\neg R(e, m, n, p)$.

We claim that $\mathbb{F}_2(LP_{q(e)}) = \emptyset$ if and only if $e \in C$. That is, suppose that $e \in C$. Then for all m, there exists an n such that for all p, $\neg R_e(m, n, p)$. Thus for all m, there is an n such that $B(\overline{m}, \overline{n})$ is not in $\mathbb{T}_0(LP_{q(e)})$ so that $A(\overline{m}) \leftarrow$ will be in $GL_{T_0(LP_{q(e)})}(LP_{q(e)})$. But then $T_1(LP_{q(e)}) = LM(GL_{T_0(LP_{q(e)})}(LP_{q(e)}))$ will contain every $A(\overline{m})$ for every m. One can then show that the clauses (iii) and (iv) will ensure that $T_1(LP_{q(e)}) = H(LP_{q(e)})$ so that $\mathbb{F}_2((LP_{q(e)}) = \emptyset$.

Suppose that $e \notin C$. Then there is an m such that $(\forall n)(\exists p) R(e, m, n, p)$. But then $B(\overline{m}, \overline{n}) \in T_0((LP_{q(e)})$ for all n so that $GL_{T_0(P_e)}(P_e)$ will not contain the rule $A(\overline{m}) \leftarrow$. It follows that $A(\overline{m}) \notin T_1((LP_{q(e)})$ and therefore $\mathbb{F}_2((LP_{q(e)}) \neq \emptyset$.

It follows that $a \in C \iff q(a) \in \{e : \mathbb{F}_2(LP_e) = \emptyset\}$. Hence every Π^0_3 predicate is many-one reducible to $\{e : \mathbb{F}_2(LP_e) = \emptyset\}$ so that $\{e : \mathbb{F}_2(LP_e) = \emptyset\}$ is Π^0_3 complete.

Our next 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., $LM(Horn(P)) \subseteq T_{wfs}(P)$. We say that the well-founded semantics of P is trivial if $T_{wfs}(P) = \widehat{\mathbb{T}}_{wfs}(P) = LM(Horn(P))$. Thus we are interested in the complexity of the set

$$I_{LP}(\text{triv-}\mathit{wfs}) = \{e : T_{wfs}(LP_e) = \widehat{\mathbb{T}}_{wfs}(LP_e) = LM(Horn(LP_e))\}. \tag{9}$$

Theorem 13 *I_{LP}(triv-wfs) is Π^0_2-complete.*

Proof Let $M_e = LM(Horn(LP_e))$. Clearly, $M_e \subseteq LM(GL_S(LP_e))$ for all $S \subseteq H(LP_e)$. Thus it follows that $T_{wfs}(LP_e) = M_e$ if and only if $\mathbb{T}_\alpha(LP_e) = M_e$ for all even α. We claim that the condition that $\widehat{\mathbb{T}}_{wfs}(LP_e) = M_e$ forces that $LM(GL_{M_e}(LP_e)) = M_e$. That is, suppose $LM(GL_{M_e}(LP_e)) = A_e \neq M_e$. Then

$$\mathbb{T}_1(LP_e) = LM(GL_{\mathbb{T}_0(LP_e)}(LP_e)) = LM(GL_{M_e}(LP_e)) = A_e$$

and

$$\mathbb{T}_2(LP_e) = LM(GL_{\mathbb{T}_1(LP_e)}(LP_e)) = LM(GL_{A_e}(LP_e)) = M_e.$$

Then one can prove by induction that $\mathbb{T}_\alpha(LP_e) = M_e$ for all even α and $\mathbb{T}_\beta(LP_e) = A_e$ for all odd β which would imply that $\widehat{\mathbb{T}}_{wfs}(LP_e) = A_e$. It thus follows that $T_{wfs}(LP_e) = \widehat{\mathbb{T}}_{wfs}(LP_e) = M_e$ if and only if $\mathbb{T}_0(LP_e) = LM(GL_{H(LP_e)}(LP_e)) = M_e$ and $LM(GL_{M_e}(LP_e)) = M_e$. However, $T_0(LP_e) = LM(GL_{H(LP_e)}(LP_e))$ is r.e. and $LM(GL_{M_e}(LP_e))$ is Σ^0_2. Now suppose that A is Σ^0_2 set and B is a Σ^0_1 and $B \subseteq A$. Then $A = B$ if and only if $\forall x(x \notin B \Rightarrow x \notin A)$ which is a Π^0_2 predicate. It follows that the condition that

$\mathbb{T}_0(LP_e) = LM(GL_{H(LP_e)}(LP_e)) = M_e$ and $LM(GL_{M_e}(LP_e)) = M_e$ are Π^0_2 predicates. Thus I_{LP}(triv-*wfs*) is Π^0_2.

To show that I_{LP}(triv-*wfs*) is Π^0_2-complete, we will use that fact that $Inf = \{e : W_e \text{ is infinite}\}$ is a Π^0_2 complete set. For any e, let

$$W_{g(e)} = \{i : |W_e| \geq i\}.$$

Let K_e be a finite predicate Horn logic program with constant symbol $\overline{0}$, unary predicate s, and unary predicate symbol C such that $C(\overline{n})$ is in the least model of K_e if and only if $n \in W_{g(e)}$. Note that K_e may have other predicates, but one can construct K_e so that only ground terms are $\overline{0}$ and $s^n(\overline{0}) = \overline{n}$.

Then there is a recursive function h such that program $LP_{h(e)}$ consists of K_e together with the following clauses:

(i) $In(x) \leftarrow C(x)$
(ii) $In(s(x)) \leftarrow \neg In(x)$
(iii) $Q(x_1, \ldots, x_k) \leftarrow In(x_1), In(x_2), \ldots, In(x_k)$ for all predicates Q of K_e which are different from C and In.

Now if W_e is infinite, then $LM(Horn(LP_{h(e)}))$ equals $H(LP_{h(e)})$ since we will be able to derive $In(\overline{n})$ and $C(\overline{n})$ for all $n \in \omega$ so that the clauses in (iii) will ensure that $LM(Horn(LP_{h(e)})) = H(LP_e)$. In that case, we can prove by induction that $\mathbb{T}_\alpha(LP_{h(e)}) = H((LP_{h(e)})$ for all $\alpha > 0$ so that $T_{wfs}(LP_{h(e)}) = \widehat{\mathbb{T}}_{wfs}(LP_{h(e)}) = LM(Horn(LP_{h(e)})$. Now suppose that W_e is finite, say $|W_e| = n$. Then it will be the case that $\mathbb{T}_0(LP_{h(e)}) = LM(Horn(LP_{h(e)}))$ restricted to the predicates $C(\cdot)$ and $In(\cdot)$ will equal $\{C(\overline{0}), I(\overline{0}), \ldots, C(\overline{n}), I(\overline{n})\}$. But then $T_1(LP_{h(e)}) = LM(GL_{\mathbb{T}_0(LP_{h(e)})}(LP_{h(e)}))$ will contain $In(\overline{m})$ for all $m > n+1$ since the clauses in (ii) will ensure that $In(\overline{m}) \leftarrow$ is in $GL_{T_0(LP_{h(e)})}(LP_{h(e)})$ for all $m > n+1$. But then $GL_{T_1(LP_{h(e)})}(LP_{h(e)})$ will contain the clause $In(\overline{n+2}) \leftarrow$ which means that $In(\overline{n+2}) \in T_2(LP_{h(e)}) \subseteq \mathbb{T}_{wfs}(LP_{h(e)})$. Thus if W_e is finite, then
$\mathbb{T}_{wfs}(LP_{h(e)}) \neq LM(Horn(LP_{h(e)})$. Thus h shows that $a \in \{e : W_e \text{ is infinite}\}$ if and only if $h(a) \in I_{LP}$(triv-*wfs*) so that I_{LP}(triv-*wfs*) is Π^0_2 complete.

5 Conclusions

In this paper, we have shown that there is a very close connection between Van Gelder's alternating fixed point algorithm to compute the well-founded semantics of a finite predicate logic program and the classical construction of the perfect kernel $K(Q)$ of a Π^0_1 class $Q \subseteq 2^\omega$ via the transfinite iteration of the Cantor-Bendixson derivative, see Theorem 5. Theorem 5 allows to transfer many complexity results concerned with index sets associated with the problem of constructing the perfect kernel $K(Q)$ of a Π^0_1 class $Q \subseteq 2^\omega$ to complexity results of index sets associated with the problem of finding the well-founded semantics of a finite predicate logic program. This allows us to not only recover the complexity results of Schlipf [S95] and Fitting [F01], but to refine their results and to prove a number of new results. In fact, many more such complexity results for index sets associated with the problems of finding the well-founded semantics of a finite predicate logic program can be proved using the same methods.

Finally, we examined the complexity of the index sets
$A = \{e : \check{\mathbb{T}}_{wfs}(LP_e) = \emptyset\}$,
$B = \{e : \widehat{\mathbb{F}}_{wfs}(LP_e) = \emptyset\}$,
$C = \{e : \mathbb{T}_{wfs}(LP_e) = \emptyset\}$,
$D = \{e : \mathbb{T}_{wfs}(LP_e) = \widehat{\mathbb{T}}_{wfs}(LP_e) = LM(Horn(LP_e)\}$, and
$E = \{e : \mathbb{F}_{wfs}(LP_e) = \emptyset\}$.
We showed that A is recursive, B, C, and D are Π^0_2 complete, and D is Π^0_3 complete.

Acknowledgments

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

[AN78] H. Andreka and I. Nemeti, The Generalized Completeness of Horn Predicate Logic as a Programming Language. *Acta Cybernetica*, 4:3–10, 1978.

[B1883] L. Bendixson, Quelques theéorèms de la th'eorie des ensumble de points, Acta Math. **2** (1883), 415-429.

[C82] D. Cenzer and R. Mauldin, On the Borel class of the derived set operator, *Bull. Soc. Math. France* **110** (1982) 357–380, II, **111**, 367–372, (1983).

[C86] D. Cenzer, P. Clote, R. Smith, R. Soare and S. Wainer, Members of countable Π^0_1 classes, *Ann. Pure and Appl. Logic* **31**, 145–163, (1986).

[CR98] D. Cenzer and J.B. Remmel. Index sets for Π^0_1 classes, *Annals of Pure and Applied Logic* **93**, 3-61 (1998).

[CR99] D. Cenzer and J.B. Remmel, Π^0_1 Classes in Mathematics. *Handbook of Recursive Mathematics*, 623-821, Elsevier (1999).

[CR08] D. Cenzer and J.B. Remmel, A connection between Cantor-Bendixson derivatives and the well-founded semantics of logic programs, ISAIM 2008, The Tenth International Symposium on Artificial Intelligence and Mathematics (Fort Lauderdale, 2008).

[CRta] D. Cenzer and J.B. Remmel, Effectively Closed Sets, ASL Lecture Notes in Logic, to appear.

[Den98] M. Denecker, The Well-founded Semantics Is the Principle of Inductive Definition, *Proceedings of the 6th European Workshop on Logics in Artificial Intelligence*, Springer LNAI 1489, pages 1-16, 1998.

[DBM01] M. Denecker and M. Bruynooghe and V.W. Marek, Logic Programming Revisited: Logic Programs as Inductive Definitions, *ACM Transactions on Computational Logic*, 2:623–654, 2001.

[F01] M.C. Fitting, Fixpoint semantics for logic programming–a survey, *Theoretical Computer Science* 278 2001, 25-51.

[GL88] M. Gelfond and V. Lifschitz, The stable semantics for logic programs, in R. Kowalski and K. Bowen, editors, ICLP88, 1070-1080 (1988)

[Hin78] P. G. Hinman, *Recursion-Theoretic Hierarchies* (Springer-Verlag, 1978).

[Kur70] K. Kuratowski, Some problems concerning semi-continuous set-valued mappings, in: *Set-Valued Mappings, Selections and Topological Properties of* 2^X, (Lecture Notes in Math., Vol. 171, Springer-Verlag, 45–48, (1970).

[L87] S. Lempp, Hyperarithmetical index sets in recursion theory, *Trans. Amer. Math. Soc.* **303**, 559–583, (1987).

[MNR94] W. Marek, A. Nerode, and J. B. Remmel, The stable models of predicate logic programs. *Journal of Logic Programming*, **21(3)**, 129-154 (1994).

[S95] J.S. Schlipf, The expressive power of the logic programming semantics, *J. Comp. and System Sciences*, **51**, 64-86, (1995).

[Soa87] R. Soare, *Recursively Enumerable Sets and Degrees* (Springer-Verlag, 1987).

[Sm68] R.M. Smullyan. *First-order Logic.* Springer-Verlag, 1968.

[T55] A. Tarski, A lattice-theoretical theorem and its applications, *Pacific J. Math.*, **5**, 285-309, (1955).

[V89] A. Van Gelder, The alternating fixpoints of logic programs with negation, *ACM Symposium on Principles of Database Systems*, 1-10, (1989).

[V93] A. Van Gelder, The alternating fixpoints of logic programs with negation, *Journal of Computer and System Sciences*, 185-221, (1993).

[VRS91] A. Van Gelder, K.A. Ross and J.S. Schlipf, The well-founded semantics for general logic programs, *J. ACM* **38(3)**, 620-650 (1991).