Nonmonotonic Rule Systems with recursive sets of restraints

V. Wiktor Marek[1]
Anil Nerode[2]
Jeffrey B. Remmel[3]

To Gerald Sacks and the Golden Age of Recursion Theory

Abstract
We study nonmonotonic rule systems with rules that admit infinitely many restraints. We concentrate on the case when the constraints of rules form a recursive sets and there is a uniform enumeration of codes for rules. We show that the theory developed for nonmonotnic rule systems admitting the rules with finite number of restraints can be lifted to such rule systems. We give tight estimates on the complexity of the set of extensions of such rule systems.

[1]Department of Computer Science, University of Kentucky, Lexington, KY 40506, Research partially supported by NSF grant IRI-9012902. E-mail: *marek@cs.uky.edu*

[2]Mathematical Sciences Institute, Cornell University, Ithaca, NY 14853, Research supported by US ARO contract DAAL03-91-C-0027. E-mail: *anil@math.cornell.edu*

[3]Department of Mathematics, University of California, La Jolla, CA 92093. Research partially supported by NSF grant DMS-9306427. E-mail: *remmel@kleene.ucsd.edu*

1 Introduction

In this paper we investigate the properties and the expressive power of structures we call "extended nonmonotonic rule systems". These structures generalize in a natural fashion nonmonotonic rule systems that we have studied extensively in the series of papers [MNR90, MNR92a, MNR94a, MNR94b, MNR94c].

Nonmonotonic rule systems arose as a simple algebraic structure capturing the essential features of what have been called "nonmonotonic logics". These logics arose in attempts to formalize several forms of commonsense reasoning. These systems include: the default logic of Reiter [Rei80], the autoepistemic logic of Moore [Moo85], the nonmonotonic modal logics of McDermott and Doyle [MD80, McD82], the truth maintenance systems of Doyle [Doy79], the stable semantics of general logic programs [GL88], the answer sets semantics for logic programs with classical negation [GL90], and the answer set semantics for clausal logic programs [YBB92].

Nonmonotonic rule systems capture in an abstract fashion all the common features of the above mentioned reasoning systems. Here is the definition. A nonmonotonic rule system is a pair $\langle U, N\rangle$, where U is any set and N is a collection of rules of the form

$$r = \frac{a_1, \ldots, a_m : b_1, \ldots, b_n}{c} \tag{1}$$

where $a_1, \ldots, a_m, b_1, \ldots, b_n, c \in U$. The elements $a_1, \ldots, a_n$, are called the *premises* of r, the elements $b_1, \ldots, b_n$ are called the *restraints* of r, the element c is called the *conclusion* of r. Each of the numbers m, n can be 0. If $n = 0$, then we call the rule r *monotonic*. If $m = n = 0$, then we call r an *axiom*.

Systems $\langle U, N\rangle$, where all the rules in N are monotonic, are well-known and well-understood. The associated notion of derivation in such systems was characterized by Tarski [Tar56], see also Birkhoff [Bir35] and Schmidt [Sch52]. In the case of *nonmonotonic* rule systems, there also is a corresponding notion of derivation. This notion of derivation is defined so as to encompass all the nonmonotonic reasoning modes discussed above. This is the notion of a S-proof where $S \subseteq U$. The set S is intended to control the applicability of the rules. That is, in order to apply a rule we must be able to derive all its premises (i.e. $a_1, \ldots, a_n$,) as in the case of monotonic rule systems, but in addition the rule r can only be applied if *none* of the restraints $b_1, \ldots, b_n$ belong to S. This notion of provability can easily be transformed into a formal inductive definition of an S-derivation. (A similar construction for extended nonmonotonic rule systems will be introduced below.) In this fashion we can define the set $C_S(I)$ of all elements of U which are S-provable from I. We then call S an **extension of** I in $\langle U, N \rangle$ if $S = C_S(I)$. Similarly, S is called an **extension** of $\langle U, N \rangle$ if S is an extension of $\emptyset$ in $\langle U, N \rangle$. It is easy to see that this concept generalizes the "bottom-up" representation (i.e. via derivable formulas) of the notion of consequence, but not the "top-down" definition of the notion of consequence as the least set closed under rules. For more on this point, see [MT93]. Moreover, unlike the situation for monotonic rule systems, there is no guarantee that a given nonmonotonic rule system $\langle U, N \rangle$ will have even one extension. It is also possible for $\langle U, N \rangle$ to have many extensions. We shall be interested in studying the family $\mathcal{E}(\langle U, N \rangle)$ of all extensions of $\langle U, N \rangle$.

If $\langle U, N \rangle$ is a nonmonotonic rule system, where U is a subset of the natural numbers ω, then one can use standard codes for finite subsets of natural numbers plus recursive pairing functions to code each rule r of the form (1) as a natural number. Then we can can identify N with a subset of ω. We then say that $\langle U, N \rangle$ is a *recursive nonmonotonic rule system* if both U and N are recursive. In general, when $U \subseteq \omega$, the family $\mathcal{E}(\langle U, N \rangle)$ of all extensions of $\langle U, N \rangle$ is a subset of the power set of ω, $\mathcal{P}(\omega)$. When $U \subseteq \omega$, Ferry [Fer93] was able to give a topological characterization of the sets of the form $\mathcal{E}(\langle U, N \rangle)$. A characterization of the expressive power of recursive nonmonotonic rule systems was given in [MNR92a]. It turns out that for every recursive nonmonotonic rule system $\langle U, N \rangle$, there exists a recursive tree $T \subseteq \omega^{<\omega}$ such that there is a one-to-one effective, degree-preserving correspondence between $\mathcal{E}(\langle U, N \rangle)$ and $[T]$, the set of all infinite paths through T. Vice versa, for every recursive tree $T \subseteq \omega^{<\omega}$, there is a recursive nonmonotonic rule system $\langle U, N \rangle$ such that there is a one-to-one effective, degree-preserving correspondence between $[T]$ and $\mathcal{E}(\langle U, N \rangle)$. Since the degrees of the elements in $[T]$ for recursive trees $T \subseteq \omega^{<\omega}$ have been extensively studied, our results allowed us to derive various results about the possible set of Turing degrees of extensions of recursive nonmonotonic rule systems and on the complexity of various decision problems associated with nonmonotonic reasoning systems mentioned above. Moreover, this says that the quotients of effective closed (Π^0_1) subsets of Cantor space by $\equiv_T$ are identical to quotients of sets of extensions of recursive nonmonotonic rule systems by the same equivalence relation.

It turns out that one can get a finer measure of the complexity on the set of extensions of a nonmonotonic rule system $\langle U, N\rangle$ by classifying nonmonotonic rule systems according to the number of so-called minimal proof schemes which are possessed by the elements of U. Briefly, a proof scheme p for an element $x \in U$ is a formal derivation of x using the rules N which simultaneously keeps track of the restraints of the rules used in the derivation. The collection of all restraints used in such a proof scheme p is called the support of the proof scheme and is denoted by $supp(p)$. Then for any S such that $S \cap supp(p) = \emptyset$, $x \in C_S(\emptyset)$. The formal definition of proof scheme was first given in [MNR92a] and will be repeated below. There is a natural well founded preordering of the set of proof schemes of x obtained by declaring that a proof scheme p is less than a proof scheme p' if every rule which occurs in p' also occurs in p. There are two natural restrictions we can put on nonmonotonic rule systems. First, one can require that every element $x \in U$ possesses only a finite number of minimal proof schemes. Such nonmonotonic rule systems are called *locally finite*. A second, more restrictive, condition on a recursive nonmonotonic rule system is that the system be locally finite and that in addition there exists a recursive function assigning to each element $x \in U$, the explicit index of the finite set of all minimal proof schemes of x. Such recursive nonmonotonic rule systems are called *highly recursive*. It is proved in [MNR92a], that for every locally finite recursive nonmonotonic rule system $\langle U, N\rangle$, there exists a finitely branching recursive tree $T \subseteq \omega^{<\omega}$ such that there is a one-to-one effective, degree-preserving correspondence between $\mathcal{E}(\langle U, N\rangle)$ and $[T]$ and, vice versa, for every finitely branching recursive tree $T \subseteq \omega^{<\omega}$, there is a locally finite recursive nonmonotonic rule system $\langle U, N\rangle$ such that there is a one-to-one effective, degree-preserving correspondence between $[T]$ and $\mathcal{E}(\langle U, N\rangle)$. Similarly, for every highly recursive nonmonotonic rule system $\langle U, N\rangle$, there exists a highly recursive tree $T \subseteq \omega^{<\omega}$ such that there is a one-to-one effective, degree-preserving correspondence between $\mathcal{E}(\langle U, N\rangle)$ and $[T]$ and, vice versa, for every highly recursive tree $T \subseteq \omega^{<\omega}$, there is a highly recursive nonmonotonic rule system $\langle U, N\rangle$ such that there is a one-to-one effective, degree-preserving correspondence between $[T]$ and

$\mathcal{E}(\langle U, N \rangle)$. Once again, since the degrees of the elements in $[T]$ for locally finite and highly recursive trees $T \subseteq \omega^{<\omega}$ have been extensively studied, our results allowed us to derive various results about the set of Turing degrees of extensions of locally finite and highly recursive nonmonotonic rule systems and, consequently, on the complexity of various decision problems associated with nonmonotonic reasoning systems mentioned above. Moreover, this says that quotients of bounded Π^0_1 subsets of Cantor space by $\equiv_T$ are identical to quotients of sets of extensions of locally finite recursive nonmonotonic rule systems by the same equivalence relation and that quotients of recursively bounded Π^0_1 subsets of Cantor space by $\equiv_T$ are identical to quotients of sets of extensions of highly recursive nonmonotonic rule systems by the same equivalence relation. See section 8 for more precise statements of these results.

Ferry [Fer91] noticed that the process of finite derivability remains the same if we permit rules similar to (1), but allowing infinitely many restraints as well as finitely many. Specifically, she considered rules of the form

$$r = \frac{a_1, \ldots, a_m : Z}{c} \tag{2}$$

where $a_1, \ldots, a_m \in U$, $Z \subseteq U$, $c \in U$. The elements $a_1, \ldots, a_n$, are called the *premises* of r, the elements of Z are called the *restraints* of r, and the element c is called the *conclusion* of r.

Nonmonotonic rule systems under prior definitions correspond to the case when Z is finite for all rules in N. Ferry proved that many properties of nonmonotonic rule systems generalize to such extended nonmonotonic rule systems. There are, however, important differences. First, when $U \subseteq \omega$, the rules with infinite Z cannot generally be coded by natural numbers. Second, it turns out that the topological characterization of the sets of the form $\mathcal{E}(\langle U, N \rangle)$ are best expressed in terms of the Ellentuck topology [Ell74] rather than the Cantor topology in $\mathcal{P}(\omega)$ or its close relative, the dual Scott topology [Fer92, Bat89].

The principal purpose of this paper is to study the expressive power of extended nonmonotonic rule systems $\langle U, N \rangle$. First we show that when U is a recursive subset of ω and each rule r of N, as described in (2), has the property that the set of restraints Z of r is a recursive set, then one can define natural analogues of recursive, locally finite, and highly recursive nonmonotonic rules systems for extended nonmonotonic rule systems. In contrast with nonmonotonic rule systems, we shall show that the set of extensions of a highly recursive *extended* nonmonotonic rule system can represent an arbitrary Π^0_1 class up to Turing equivalence. That is, we shall show that for any recursive tree $T \subseteq \omega^{<\omega}$, there is a highly recursive extended nonmonotonic rule system $\langle U, N \rangle$ such that there is a one-to-one effective, degree-preserving correspondence between $[T]$ and $\mathcal{E}(\langle U, N \rangle)$. We only have a weaker version of the converse result. That is, for any recursive extended nonmonotonic rule system $\mathcal{S}$, we construct a recursive tree T and a one-to-one correspondence between the extensions of $\mathcal{S}$ and the branches through T. Although the extension is uniformly recursive in the corresponding branch, the branch is only recursive in the Turing jump of the extension. This result still allows us to get some natural results on the degrees of extensions of extended nonmonotonic rule systems. For instance, it follows that if an extended recursive nonmonotonic rule system possesses a unique extension, then this extension is hyperarithmetic. At the end of Section 9 we show that a stronger version of the correspondence result is false. Specifically, using a wait-and-see technique we construct an example of an extended nonmonotonic rule system for which there is no recursive specification of the one-to-one correspondence described above. The example (although not the argument) was suggested by the anonymous referee of the arlier version of this paper.

This paper continues our program of demonstrating a close connection between the nonmonotonic reasoning systems introduced by research workers in Artificial Intelligence and Classical Recursion Theory. The ubiquitous relationship between Logic and Recursion Theory persists in this context. The new modes of nonmonotonic reasoning currently considered in Artificial Intelligence can be fruitfully analyzed by recursion-theoretic methods.

Finally, we express our gratitude to the referee of the earlier version of this paper for both corrections to version s/he read as well as the suggestion for the counterexample to the strengthening of Theorem 9.9.

2 Examples of Nonmonotonic logical systems

We will introduce briefly some of the nonmonotonic logical systems mentioned in the introduction. We introduce three such systems: the default logic of Reiter [Rei80], the nonmonotonic modal logics of McDermott [McD82], and general logic programming [Apt90]. It will be shown later that each of these example can be coded into nonmonotonic rule systems. Moreover, because the coding of such systems into nonmonotonic rules systems is so direct, it will be clear that there are natural "extended" analogues of such systems and that all the results on extended nonmonotonic rule systems proved in this paper immediately apply to such extended systems.

2.1 Default Logic

Default Logic is a system based on the language of propositional logic and is able to handle those rules which, informally, have the format: "normally, if α is proven, then derive γ". Here the word "normally" is interpreted as the "lack of any indication of abnormality". To make this more precise, default logic handles default rules, which are entities in the following format:

$$r = \frac{\alpha : M\beta_1, \ldots, M\beta_k}{\gamma}$$

where $\alpha, \beta_1, \ldots, \beta_k, \gamma$ are formulas of the underlying propositional language $\mathcal{L}$. Thus, in this case, normality is interpreted as the fact that there is "no information that $\neg\beta_1$, . . . , $\neg\beta_k$ holds". Following Reiter [Rei80], a default theory is defined as a pair $\langle D, W\rangle$ where D is a set of default rules and W is a set of formulas of $\mathcal{L}$.

We define an operator Γ mapping $P(\mathcal{L})$, the set of all subsets of the well formed formulas of $\mathcal{L}$, into itself as follows: $\Gamma(S)$ is the least set U satisfying the following three properties:

1. $W \subseteq U$
2. $Cn(U) = U$
3. For all rules $r \in D$, if $\alpha \in U$ and $\neg\beta_1 \notin S \ldots, \neg\beta_k \notin S$ then $\gamma \in U$.

Here $Cn(U)$ denotes the set of all logical consequences of U. It is easy to see that the operator Γ is well-defined. We then call a set of formulas S an *extension* of $\langle D, W\rangle$ if $S = \Gamma(S)$.

Default logic and the extensions of default theories have been studied extensively in the Artificial Intelligence community for the past decade. It has been argued that default rules provide a mechanism to formalize many important aspects of common-sense reasoning where one naturally makes conclusions which are based not only the presence of information but also on the current absence of other information.

2.2 Nonmonotonic modal logics

McDermott [McD82] introduced a technique which allows one to create non-monotonic counterparts of various modal logics.

We shall use the modal language $\mathcal{L}_L$ with one modal operator L, interpreted as necessitation, knowledge, or belief. Given a modal logic $\mathcal{S}$, we consider the consequence operator $Cn_{\mathcal{S}}$. This consequence operator is different from one used in most work on modal logic (see Chellas [Che80]) in that it allows for application of necessitation to all previously proven formulas, not only to the the axioms of $\mathcal{S}$. McDermott studied the proper form of Kripke semantics for such notion of provability.

Now, given a set of formulas $T \subseteq \mathcal{L}$ and another set of formulas I, interpreted to be the initial assumptions of the reasoning agent, we consider the set of formulas

$$Cn_{\mathcal{S}}(I \cup \{\neg L\psi : \psi \notin T\})$$

A theory T is then called an **$\mathcal{S}$-expansion of I** if

$$T = Cn_{\mathcal{S}}(I \cup \{\neg L\psi : \psi \notin T\})$$

The set of formulas $\{\neg L\psi : \psi \notin T\}$ represents the so-called "negative introspection with respect to T". Informally, this set consists of those formulas expressing the statements: "ψ is not believed", for all ψ that are not in the belief set of the reasoning agent.

Again, McDermott [McD82] argued that expansions can formalize essential aspects of common-sense reasoning. We shall see below that nonmonotonic rule systems faithfully represent McDermott expansions.

2.3 General Logic Programming

General logic programs extend the usual syntax of (Horn) logic programs by admitting negated atoms in the body of clauses. Specifically, a general clause is an expression of the form:

$$p \leftarrow q_1, \ldots, q_m, \neg r_1, \ldots, \neg r_n$$

Here we only assume that $m \geq 0$ and $n \geq 0$ so that the usual logic programming clauses are special cases of general clauses. General clauses possess the *logical interpretation*:

$$q_1 \wedge \ldots \wedge q_m \wedge \neg r_1 \wedge \ldots \wedge \neg r_n \supset p$$

As long as we are interested in Herbrand models of general programs, we can consider the propositional theory $ground(P)$ consisting of all ground substitutions of clauses of P. While P is usually finite, $ground(P)$ may be infinite if P contains function symbols. There is of course a one-to-one correspondence between Herbrand models of P and propositional models of $ground(P)$.

As is the case for (Horn) logic programming, not every model of the program has a clear computational meaning. Some models of a general program provide a computationally sound meaning to negation. Three of the most widely used used semantics are the supported models of Clark [Cla78] (see also Apt and van Emden, [AvE82]), the stable models of Gelfond and Lifschitz [GL88], and the well-founded model of Van Gelder, Ross and Schlipf [VGRS91]. We shall discuss here only the stable models of Gelfond and Lifschitz since stable models are most naturally modeled by extensions of nonmonotonic rule systems.

Since we are interested in Herbrand models, we consider the propositional program $ground(P)$. Given a subset of Herbrand universe, M, and a clause

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

in *ground*(P), we define C^M as **nil** if $r_j \in M$ for some $1 \leq j \leq n$. Otherwise $C^M = p \leftarrow q_1, \ldots, q_m$. Then we put

$$P^M = \{C^M : C \in \mathit{ground}(P)\}$$

Since P^M is a Horn program, it possesses a least Herbrand model, N_M. Then we call M a *stable model of* P if $M = N_M$. It is easy to see that a stable model of P is a model of P. We shall see that nonmonotonic rule systems allow for faithful representation of stable models of general logic programs. Consequently the results about extensions of nonmonotonic rule systems immediately imply corresponding results about stable models of logic programs.

3 Monotonic formal systems

Tarski [Tar56] characterized monotonic formal systems by means of monotonic rules of inference. Such systems include classical propositional and first order logics, intuitionistic logic, modal logics, and many others. Suppose that a nonempty set U is given. In a particular application U may be the collection of all statements or formulas, all legal strings of a formal system, or of all atomic statements as in logic programming.

Definition 3.1

1. A *monotonic rule of inference* over U is a tuple $r = \langle P, c\rangle$, where $P = \langle a_1, \ldots, a_n\rangle$ is a finite list of objects from U and c is an element of U. Such a rule r is usually written in the more suggestive form
$$r = \frac{a_1, \ldots, a_n}{c}. \tag{3}$$
We call $a_1, \ldots, a_n$ the premises of r and c the conclusion of r.

2. A monotonic rule system is a pair $\langle U, M\rangle$, where U is a nonempty set and M is a collection of monotonic rules.

3. A subset $S \subseteq U$ is called *deductively closed* over $\langle U, M\rangle$ if for all rules $r \in M$ as in (3), $a_1, \ldots, a_n \in S$ implies $c \in S$.

The collection $D = D(U, M)$ of deductively closed sets has the following properties:

1. $U \in D$.
2. D is closed under arbitrary intersections. (Consequently, for every $I \subseteq U$ there is the least set $T(I)$ such that $I \subseteq T(I)$ and $T(I)$ is deductively closed. The operator $T(= T_{U,M})$ is monotone, that is, if $I \subseteq J$ then $T(I) \subseteq T(J)$). Moreover,
3. $T(I) = \bigcup\{T(J) \colon J \subseteq I \ \&|J| < \omega\}$.

Property (3) reflects the finitary nature of deductive closure and is closely associated with the definition of a deduction.

An *axiom* is a rule without premises, that is, with the list P empty.

A *deduction* of an object $c \in U$ from $I \subseteq U$ is a finite sequence $\langle c_1, \ldots, c_m \rangle$ such that $c_m = c$, and for all $i \leq m$, $c_i \in I$, or c_i is an axiom, or c_i is the conclusion of a rule $r \in M$ such that premises of r are included in $\{c_1, \ldots, c_{i-1}\}$. Then $T(I)$ consists of all elements of U that possess a deduction from I.

Tarski noticed that if a collection D of subsets of U possesses properties (1), (2) and (3) above, then there is a collection of monotone rules M such that D is the set of all deductively closed sets in $\langle U, M \rangle$.

An abstract treatment of monotonic logic programming schemes and general methods of processing queries is given in [BBS89].

4 Nonmonotonic formal systems

Inspired by Reiter [Rei80], and Apt [Apt90], we introduced in [MNR90] the notion of a nonmonotonic formal system $\langle U, N \rangle$.

Definition 4.1 1. A nonmonotonic rule of inference is a triple $\langle P, G, c \rangle$, where $P = \{a_1, \ldots, a_n\}$, $G = \{b_1, \ldots, b_m\}$ are finite lists of objects from U and $c \in U$. Each such rule is written in form

$$r = \frac{a_1, \ldots, a_n \colon b_1, \ldots, b_m}{c} \tag{4}$$

Here $a_1, \ldots, a_n$ are called the *premises* of rule r, $b_1, \ldots, b_m$ are called the *restraints* of rule r. Either P, or G, or both may be empty. If $P = G = \emptyset$, then the rule r is called an axiom. The set $\{a_1, \ldots, a_n\}$ is denoted by $p(r)$, the set $\{b_1, \ldots, b_m\}$ is denoted by $R(r)$, and c is denoted by $c(r)$.

2. A *nonmonotonic rule system* is a pair $\langle U, N \rangle$ where U is a non-empty set and N is a set of nonmonotonic rules.

Each monotonic rule system can be identified with the nonmonotonic rule system in which every monotonic rule is given an empty set of restraints.

A subset $S \subseteq U$ is called *deductively closed* if for every rule r of N, whenever all premises $a_1, \ldots, a_n$ are in S and all restraints $b_1, \ldots, b_m$ are not in S, then the conclusion c belongs to S.

In nonmonotonic systems, deductively closed sets are not generally closed under arbitrary intersections as in the monotone case. Tarski's axioms do not generally hold. But deductively closed sets are closed under intersections of *descending* chains. Since U is deductively closed, by the Kuratowski-Zorn Lemma, any $I \subseteq U$ is contained in at least one *minimal* deductively closed set.

Example 4.1 *Let $U = \{a, b, \gamma\}$.*
(a) Consider U with $N_1 = \{\frac{:}{a}, \frac{a:b}{b}\}$. There is only one minimal deductively closed set $S = \{a, b\}$.
(b) Consider U with $N_2 = \{\frac{:}{a}, \frac{a:b}{\gamma}, \frac{a:\gamma}{b}\}$,
then there are two minimal deductively closed sets, $S_1 = \{a, b\}, S_2 = \{a, \gamma\}$.

Definition 4.2 1. Given a set S and an $I \subseteq U$, an S-deduction of c from I in $\langle U, N \rangle$ is a finite sequence $< c_1, \ldots, c_k >$ such that $c_k = c$ and, for all $i \leq k$, each c_i is in I, or is an axiom, or is the conclusion of a rule $r \in N$ such that all the premises of r are included in $\{c_1, \ldots, c_{i-1}\}$ and all restraints of r are in $U \setminus S$ (see [MT89a], also [RDB89]).

2. An S-consequence of I is an element of U occurring in some S-deduction from I.

3. $C_S(I)$ is the set of all S-consequences of I in $\langle U, N\rangle$.

Note that I is a subset of $C_S(I)$ and that S enters solely as a restraint on the use of the rules imposed by the restraints in the rules. A single restraint in a rule in N may be in S and therefore prevent the rule from ever being applied in an S-deduction from I, even though all the premises of that rule occur earlier in the deduction. Thus S contributes no members directly to $C_S(I)$, although members of S may turn up in $C_S(I)$ by an application of a rule which happens to have its conclusion in S. For a fixed S, the operator $C_S(\cdot)$ is monotonic. That is, if $I \subseteq J$, then $C_S(I) \subseteq C_S(J)$. Also, $C_S(C_S(I)) = C_S(I)$. However, $C_S(\cdot)$ is antimonotonic in S, that is for fixed I, $S' \subseteq S$ implies that $C_S(I) \subseteq C_{S'}(I)$.

Generally, $C_S(I)$ is not deductively closed in $\langle U, N\rangle$. It is perfectly possible to have all the premises of a rule be in $C_S(I)$, all the restraints of that rule be outside $C_S(I)$, but a restraint of that rule be in S, preventing the conclusion from being put into $C_S(I)$.

Example 4.2 $U = \{a, b, \gamma\}, N = \{\frac{:}{a}, \frac{a:b}{\gamma}\}$, $S = \{b\}$. *Then* $C_S(\emptyset) = \{a\}$ *is not deductively closed.*

We list below some basic properties of extensions (recall that an extension is a solution to the equation $C_S(\emptyset) = S$).

Proposition 4.3 *1. If S is an extension of I, then:*

(a) S is a minimal deductively closed superset of I.

(b) For every I' such that $I \subseteq I' \subseteq S$, $C_S(I') = S$.

2. The set of extensions of I forms an antichain. That is, if S_1, S_2 are extensions of I and $S_1 \subseteq S_2$, then $S_1 = S_2$.

Given $S \subseteq U$, a rule r is called S-*applicable* if all the restraints of r are outside S and all the premises of r are in S. We define $N(S)$ to be the collection of all S-applicable rules.

With each rule r of form (4), we associate a monotonic rule of form (3)

$$r' = \frac{a_1, \ldots, a_n}{c} \tag{5}$$

obtained from r by dropping all the restraints. The rule r' is called the *projection* of rule r. We write $M(S)$ for the collection of all projections of all rules from $N(S)$. The projection $\langle U, N \rangle \mid_S$ is the monotonic rule system $\langle U, M(S) \rangle$. Thus $\langle U, N \rangle \mid_S$ is obtained as follows: First, non-S-applicable rules are eliminated. Then, the restraints are dropped altogether. We have the following characterization theorem.

Theorem 4.4 *A subset $S \subseteq U$ is an extension of I in $\langle U, N \rangle$ if and only if S is the deductive closure of I in $\langle U, N \rangle \mid_S$.*

Theorem 4.4 tells us how to test if a collection $S \subseteq U$ is an extension of I in $\langle U, N \rangle$. In case U and N are finite, this leads to an implementable algorithm.
(1) Compute $N(S)$.
(2) Project $N(S)$ by dropping restraints to get $M(S)$.
(3) Compute the deductive closure of I in $\langle U, M(S) \rangle$, call this T.
(4) Test whether $T = S$.

Note that if U is finite, then it is easy to see that one can test whether S is an extension in linear time over the input $\langle U, N \rangle$. Finding all the extensions of a given I is a more complex problem. A brute force algorithm is to generate all subsets of U, and test each of them for being an extension using the procedure above. A useful fact for improving this algorithm is the following.

Proposition 4.5 *If S is a extension of I, then S consists entirely of elements of I and conclusions of certain rules in N.*

Proposition 4.5 allows us to possibly cut down on the number of subsets that we have to check to find all extension. Nevertheless, we are still left with an exponential time algorithm to find all the extensions of $\langle U, N \rangle$. It is unlikely this can be improved in general. For example, it is known that the problem of determining whether a given nonmonotonic rule system has an extension is NP complete, the problem of determining whether a given

$x \in U$ is in some extension of $\langle U, N \rangle$ is NP complete, and the problem of determining whether a given $x \in U$ is all extensions is co-NP complete [MT91, BF91].

A simple construction allows us to consider only extensions of the empty set. In fact, if $\mathcal{S}$ is a nonmonotonic rule system, and $I \subseteq U$, then the system $\mathcal{S}(I)$ arises from $\mathcal{S}$ and I by adding to N all the rules of the form $\frac{:}{t}$ for all $t \in I$. We then have:

Proposition 4.6 *T is an extension of I in $\mathcal{S}$ if and only if T is an extension of $\emptyset$ in $\mathcal{S}(I)$.*

Most systems of logic can be represented as a monotonic rule system. In [MNR90] we provided the details of the representation of such formalisms as rule systems. It is rather straightforward that the following systems can be represented a *monotonic* rule systems: classical propositional logic, intuitionistic propositional logic, classical modal systems. The fragment of propositional logic, Horn logic, which forms a logical basis for logic programming also admits such representation.

As pointed our in the introduction, the formalism of *nonmonotonic* rule systems also provides a uniform description of several systems currently considered in the literature of artificial intelligence. These include: default logic of Reiter [Rei80], general logic programming (i.e. propositional logic programming with negation as failure and stable semantics, [GL88]), logic programming with classical negation, [GL90], logic programming with clauses [YBB92], nonmonotonic modal logics of McDermott [MD80, McD82], autoepistemic logic of Moore [Moo85] and truth maintenance system of Doyle [Doy79]. In the next section, we will briefly describe how the systems described in section 2 correspond to nonmonotonic rule systems.

In short, the results on nonmonotonic rule systems carry over to all these domains. This implies that the nonmonotonic rule systems provide a uniform method of getting results in all the application areas mentioned above.

5 Encoding various nonmonotonic logics by means of rule systems

In this section, we show how to encode the three nonmonotonic logics discussed in Section 2 as nonmonotonic rule systems.

5.1 Encoding default logic

Let $\mathcal{L}$ be the propositional language underlying the given default logic. With $\mathcal{L}$ fixed, all our nonmonotonic rule systems will have the same universe, namely the set of all well-formed formulas of $\mathcal{L}$. We now show how to interpret a given default theory as a nonmonotonic rule system.

Let $\langle D, W \rangle$ be a default theory. For every default rule r,

$$r = \frac{\alpha : M\beta_1, \ldots, M\beta_k}{\gamma}$$

construct the following nonmonotonic rule d_r

$$d_r = \frac{\alpha : \neg\beta_1, \ldots, \neg\beta_k}{\gamma}$$

Next, for every formula $\psi \in \mathcal{L}$, define the rule

$$d_\psi = \frac{:}{\psi}$$

and for all pairs of formulas χ_1, χ_2 define

$$mp_{\chi_1,\chi_2} = \frac{\chi_1, \chi_1 \supset \chi_2 :}{\chi_2}$$

Now define the set of rules $N_{D,W}$ as follows:

$$N_{D,W} = \{d_r : r \in D\} \cup \{d_\psi : \psi \in W \text{ or } \psi \text{ is a tautology}\} \cup \{mp_{\chi_1,\chi_2} : \chi_1, \chi_2 \in \mathcal{L}\}$$

We have the following result [MNR90]

Theorem 5.1 *Let $\langle D, W\rangle$ be a default theory. Then a set of formulas S is a default extension of $\langle D, W\rangle$ if and only if S is an extension of nonmonotonic rule system $\langle U, N_{D,W}\rangle$.*

Theorem 5.1 says that at a cost of a simple syntactic transformation and additional encoding of logic as (monotonic) rules, we can faithfully represent default logics by means of nonmonotonic rule systems.

5.2 Encoding modal nonmonotonic logics

Let $\mathcal{L}_L$ be a fixed language of propositional logic with an additional modal operator L. Our universe U will be as before the set of all well formed formulas of the language $\mathcal{L}_L$. In our coding, we will employ a technique similar to one used in the case of default logic.

Now, for every formula $\psi \in \mathcal{L}_L$ we consider a rule

$$e_\psi = \frac{:\psi}{\neg L\psi}$$

Next we encode the necessitation rules of $\mathcal{L}_L$. For every formula ψ we have the rule

$$n_\psi = \frac{\psi :}{L\psi}$$

Finally, like in 5.1, $d_\psi = \frac{:}{\psi}$.
Now, given a theory I (the set of initial assumptions) in a modal logic $\mathcal{S}$, consider the following set of rules $N_{I,\mathcal{S}}$

$$\begin{aligned} N_{I,\mathcal{S}} \quad = \quad & \{d_\psi : \psi \in \mathcal{S}\} \cup \{e_\psi : \psi \in \mathcal{L}_L\} \cup \\ & \{mp_{\chi_1,\chi_2} : \chi_1, \chi_2 \in \mathcal{L}_L\} \cup \{n_\psi : \psi \in \mathcal{L}_L\} \\ & \{d_\psi : \psi \in I\} \cup \{d_\psi : \psi \text{ is a tautology of } \mathcal{L}_L\} \end{aligned}$$

We then have the following result.

Theorem 5.2 *Let $I, T \subset \mathcal{L}_L$. Let $\mathcal{S}$ be a modal logic. Then T is an $\mathcal{S}$-expansion of I if and only if T is an extension of the nonmonotonic rule system $\langle U, N_{I,\mathcal{S}}\rangle$.*

Theorem 5.2 tells us that the nonmonotonic modal logic can be faithfully simulated within the formalism of nonmonotonic rule systems.

5.3 Encoding stable semantics for general logic programs

We shall encode now the stable models of logic programs as extensions of suitably chosen nonmonotonic rule systems. The universe of all our systems, U, will be the Herbrand base of the program. Next, to every general propositional clause C,

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

assign the rule

$$r_C = \frac{q_1, \ldots, q_m : r_1, \ldots, r_n}{p}$$

Now, given the program P, define

$$N_P = \{r_C : C \in \mathit{ground}(P)\}$$

We then have the following result

Theorem 5.3 *Let P be a general logic program. Let M be a subset of the Herbrand base of P. Then M is a stable model of P is and only if M is an extension of the nonmonotonic rule system $\langle U, N_P \rangle$.*

Theorem 5.3 allows us to obtain results concerning stable models of logic programs from theorems about extensions of nonmonotonic rule systems.

These theorems and other similar results obtained in [MNR90, MNR92a] indicate why nonmonotonic rule systems are useful in the investigations of various nonmonotonic logical formalisms. The results, both positive and, to some extent, negative, on nonmonotonic rule systems provide us with corresponding results for all these formalisms.

6 Proof schemes: proof theory for nonmonotonic rule systems

We define now the basic notion used to analyze the Turing complexity of the set of extensions of nonmonotonic rule systems. This is the notion of a proof scheme. Intuitively, a proof scheme is a derivation together with two additional sets of items. First, the collection of rules used in derivation. Second, the set of elements that needs to be disjoint from the set S in order to make our derivation an S-derivation.

More formally, we have:

Definition 6.1 A proof scheme s is a finite sequence of triples

$$\langle\langle c_1, r_1, Z_1\rangle, \ldots, \langle c_n, r_n, Z_n\rangle\rangle$$

where $c_1, \ldots, c_n \in U$, $r_1, \ldots, r_n \in N$, $Z_1, \ldots, Z_n$ are finite subsets of U such that for all j, $1 \leq j \leq n$

1. $c_1 = c(r_1)$, $Z_1 = R(r_1)$ and $p(r_1) = \emptyset$
2. For $j > 1$, $p(r_j) \subseteq \{c_1, \ldots, c_{j-1}\}$, $c(r_j) = c_j$, and $Z_j = Z_{j-1} \cup R(r_j)$.
3. c_n is the conclusion of s and is denoted by $cln(s)$. Z_n is called the support of s and is denoted by $supp(s)$.

Clearly an initial segment of a proof scheme is also a proof scheme.

Notice that the support of a proof scheme s, Z_n, has the property that for every set S such that $S \cap Z_n = \emptyset$, the sequence $\langle c_1, \ldots, c_n\rangle$ is an S-derivation. Conversely if $\langle c_1, \ldots, c_n\rangle$ is an S-derivation, then there is a proof scheme

$$s = \langle\langle c_1, r_1, Z_1\rangle, \ldots, \langle c_n, r_n, Z_n\rangle\rangle$$

such that $Z_n \cap S = \emptyset$.

There is a natural preordering on proof schemes. Namely, $s_1 \prec s_2$ if and only if every rule appearing in s_1 appears in s_2 as well. The relation $\prec$ is not a partial ordering but it is well-founded. We can thus talk about minimal proof schemes for a given element $c \in U$. Intuitively, a minimal proof scheme carries the minimal information necessary to derive its conclusion. Since the support of every proof scheme is finite, the negative information carried in such a proof scheme is finite.

Proof schemes can be used to characterize extensions. We say that a set S *admits* a proof scheme s if $supp(s) \cap S = \emptyset$. We then have the following characterization of extensions.

Proposition 6.2 *Let $\mathcal{S} = \langle U, N \rangle$ be a nonmonotonic rule system. Let $S \subseteq U$. Then S is an extension of $\mathcal{S}$ if and only if the following conditions are met:*

1. *If s is a proof scheme and S admits s, then $c(s) \in S$.*
2. *Whenever $a \in S$ then there exists a proof scheme s such that S admits s.*

It is easy to see that we can restrict to minimal proof schemes in Proposition 6.2.

7 Extended nonmonotonic rule systems

In this section, we shall formally define the notion of extended nonmonotonic rule systems, i.e nonmonotonic rule systems with rules that have possibly infinite sets of restraints, and investigate how much of the theory outlined above can be lifted to the present case.

Definition 7.1

1. A extended nonmonotonic rule is a triple $\langle P, Z, c \rangle$ where P is a finite subset of U, Z is a subset of U (it can be finite or infinite), and $c \in U$. When $P = \{a_1, \ldots, a_k\}$, $Z = \{b_1, \ldots\}$, then such rule is written as
$$\frac{a_1, \ldots, a_k : b_1, \ldots}{c}$$

2. A extended nonmonotonic rule system (ENRS) is a pair $\langle U, N\rangle$ where U is a set, and N consists of extended nonmonotonic rules.

Notice that this notion is a generalization of nonmonotonic rule system considered in previous sections. That is, we allow for rules having infinite sets of restraints. Indeed, if the set of restraints for any rule of N is finite, then $\langle U, N\rangle$ is a nonmonotonic rule system.

Next, observe that the concept of an S-derivation can be generalized without change to the present context. The S-derivations are finite sequence of elements of U. With a small abuse of notation we again denote the set of all elements with an S-derivation from I by $C_S(I)$.

Definition 7.2 Let $\mathcal{S} = \langle U, N\rangle$ be an extended nonmonotonic rule system.

1. A subset $S \subseteq U$ is called an extension of I if $S = C_S(I)$.
2. $\mathcal{E}(\mathcal{S})$ is the set of all extensions of $\emptyset$ in $\mathcal{S}$.

Basic properties of extensions, discussed in Section 4, lift to the present context.

The notion of proof scheme also generalizes to the case of extended nonmonotonic rule systems. This includes the characterization theorem (Proposition 6.2).

There is, however, an important difference. In the case of nonmonotonic rule systems in which all the rules have finite restraints, every proof scheme is a "finite object" over U. To make this remark more precise, treating every element of U as an urelement we find that every proof scheme is a hereditarily finite set over U. This is no longer the case when we allow for infinite restraints. Now, in principle, the proof schemes are no longer hereditarily finite over U. The reason is that the sets $R(r_j)$ do not need to have "finite description". In the next section we shall see that if for all rules $r \in N$ have the property that the set of restraints of r, $R(r)$, is a recursive set, then we can introduce a surrogate of proof scheme as a finite object. This will require a more sophisticated coding than is necessary in case of (ordinary) nonmonotonic rule systems.

8 Review of complexity results for extensions of nonmonotonic rule systems

As stated in the introduction, the main goal of this paper is to study the complexity of the family of extensions of extended nonmonotonic rule systems. However before we prove the main results of the paper, we shall briefly review what is known about the complexity of the family of extensions for nonmonotonic rule systems.

8.1 Preliminaries

Let ω denote the set of natural numbers. Let $[,]\colon \omega \times \omega \to \omega$ be a fixed one-to-one and onto recursive pairing function such that the projection functions π_1 and π_2 defined by $\pi_1([x,y]) = x$ and $\pi_2([x,y]) = y$ are also recursive. The canonical index, $can(X)$, of the finite set $X = \{x_1 < \ldots < x_n\} \subseteq \omega$ is defined as $2^{x_1} + \ldots + 2^{x_n}$ and the canonical index of $\emptyset$ is defined as 0. Let D_k be the finite set whose canonical index is k, i.e., $can(D_k) = k$.

Let $\langle U, N\rangle$ be a nonmonotonic rule system where $U \subseteq \omega$. We shall identify a rule r with the code of a triple $[k, l, \varphi]$ where $D_k = p(r)$, and $D_l = R(r)$, $\varphi = c(r)$. In this way we can think about N as a subset of ω as well. This given, we then say that a NRS $\mathcal{S} = \langle U, N\rangle$ is **recursive** if U and N are recursive subsets of ω

Next we shall define various types of recursive trees and Π^0_1 classes. We extend our recursive pairing function to code n-tuples for $n > 2$ by the usual inductive definition, that is, let $[x_1, \ldots, x_n] = [x_1, [x_2, \ldots, x_n]]$ for $n \geq 3$. Let $\omega^{<\omega}$ be the set of all finite sequences from ω and let $2^{<\omega}$ be the set of all finite sequences of 0's and 1's. Given $\alpha = \langle \alpha_1, \ldots, \alpha_n\rangle$ and $\beta = \langle \beta_1, \ldots, \beta_k\rangle$ in $\omega^{<\omega}$, write $\alpha \sqsubseteq \beta$ if α is initial segment of β, i.e. , if $n \leq k$ and $\alpha_i = \beta_i$ for $i \leq n$. In this paper, we identify each finite sequence $\alpha = \langle \alpha_1, \ldots, \alpha_n\rangle$ with its code $c(\alpha) = [n, [\alpha_1, \ldots, \alpha_n]]$ in ω. Let 0 be the code of the empty sequence $\emptyset$. When we say that a set $S \subseteq \omega^{<\omega}$ is recursive, recursively enumerable, etc., what we mean is that the set $\{c(\alpha)\colon \alpha \in S\}$ is recursive, recursively enumerable, etc. Define a **tree** T to be 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 provided that for all n, $\langle f(0), \ldots, f(n)\rangle \in T$. Let $[T]$ be the set of all infinite paths through T. A set A of functions is a Π^0_1-class if there exists a recursive predicate R such that $A = \{f\colon \omega \to \omega : \forall n(R(n, [f(0), \ldots, f(n)])\}$. We say a Π^0_1-class A is **recursively bounded** if there exists a recursive function $g\colon \omega \to \omega$ such that $\forall f \in A \forall n(f(n) \leq g(n))$. It is not difficult to see that if A is a Π^0_1-class, then $A = [T]$ for some recursive tree $T \subseteq \omega^{<\omega}$. Say that a tree $T \subseteq \omega^{<\omega}$ is **highly recursive** if T is a recursive finitely branching tree and also there is a recursive procedure which, applied to the code of a $\alpha = \langle \alpha_1, \ldots, \alpha_n\rangle$ in T, produces a canonical index of the set of of codes of the immediate successors of α in T. If A is a recursively bounded Π^0_1-class, it is easy to show that $A = [T]$ for some highly recursive tree $T \subseteq \omega^{<\omega}$, see [JS72b]. For any set $A \subseteq \omega$, let $A' = \{e\colon \{e\}^A(e) \text{ is defined}\}$ be the jump of A, let $\mathbf{0}'$ denote the jump of the empty set $\emptyset$. We write $A \leq_T B$ if A is Turing reducible to B and $A \equiv_T B$ if $A \leq_T B$ and $B \leq_T A$.

We say that there is an effective, one-to-one degree preserving correspondence between the set of extensions $\mathcal{E}(\mathcal{S})$ of a recursive (extended) nonmonotonic rule system $\mathcal{S} = \langle U, N\rangle$ and the set of infinite paths $[T]$ through a recursive tree T if there are indices e_1 and e_2 of oracle Turing machines such that
(i) $\forall f \in [T]\{e_1\}^{gr(f)} = E_f \in \mathcal{E}(\mathcal{S})$,
(ii) $\forall E \in \mathcal{E}(\mathcal{S})\{e_2\}^E = f_E \in [T]$, and
(iii) $\forall f \in [T]\forall E \in \mathcal{E}(\mathcal{S})(\{e_1\}^{gr(f)} = E$ if and only if $\{e_2\}^E = f)$.
where $\{e\}^B$ denotes the function computed by the e^{th} oracle machine with oracle B. Also, write $\{e\}^B = A$ for a set A if $\{e\}^B$ is a characteristic function of A. For any function $f\colon \omega \to \omega$, $gr(f) = \{[x, f(x)]\colon x \in \omega\}$. Condition (i) says that the infinite paths of the tree T uniformly produce extensions via an algorithm with index e_1. Condition (ii) says that extensions of $\mathcal{S}$ uniformly produce infinite paths through T via an algorithm with index e_2. Condition (iii) asserts that if $\{e_1\}^{gr(f)} = E_f$, then f is Turing equivalent to E_f. In the sequel we shall not explicitly construct the indices e_1 and e_2, but it will be clear that such indices can be constructed in each case.

There are two important subclasses of recursive NRS's introduced in [MNR92b], namely locally finite and highly recursive nonmonotonic rules systems. We say that the system $\langle U, N\rangle$ is **locally finite** if for each $c \in U$, there are only finitely many $\prec$-minimal proof schemes with conclusion c. Given a proof scheme for c, $s = \langle\langle c_1, r_1, Z_1\rangle, \ldots, \langle c_n, r_n, Z_n\rangle\rangle$, the code of s,

$c(s)$, is defined by

$$c(s) = [n, [[c_1, r_1, Z_1], \dots, [c_n, r_n, Z_n]]].$$

If $\langle U, N \rangle$ is a locally finite recursive nonmonotonic rule system and $c \in U$, we let Dr_c denote the set of codes of all $\prec$-minimal proof schemes for c. We say that $\langle U, N \rangle$ is **highly recursive** if $\langle U, N \rangle$ is recursive, locally finite, and the map $c \mapsto can(Dr_c)$ is partial recursive. The latter means that there is an effective procedure which, when applied to any $c \in U$, produces a canonical index of the set of all codes of $\prec$-minimal proof schemes with conclusion c.

8.2 Complexity results for extensions of recursive nonmonotonic rule systems

We can now state some basic results from [MNR90, MNR92a, MNR92b, MNR94b] on the complexity of extensions in recursive nonmonotonic rule systems.

Theorem 8.1 *For any highly recursive NRS system $\mathcal{S} = \langle U, N \rangle$, there is a highly recursive tree $T_{\mathcal{S}}$ such that there is an effective 1:1 degree preserving correspondence between $[T_{\mathcal{S}}]$ and $\mathcal{E}(\mathcal{S})$. Vice versa, for any highly recursive tree T, there is a highly recursive NRS system $\mathcal{S}_T = \langle U, N \rangle$ such that there is an effective 1:1 degree preserving correspondence between $[T]$ and $\mathcal{E}(\mathcal{S}_T)$.*

Theorem 8.2 *For any locally finite recursive NRS system $\mathcal{S} = \langle U, N \rangle$, there is a finitely branching recursive tree $T_{\mathcal{S}}$ such that there is an effective 1:1 degree preserving correspondence between $[T_{\mathcal{S}}]$ and $\mathcal{E}(\mathcal{S})$. Vice versa, for any highly recursive tree T in $\mathbf{0}'$, there is a locally finite recursive NRS system $\mathcal{S}_T = \langle U, N \rangle$ such that there is an effective 1:1 degree preserving correspondence between $[T]$ and $\mathcal{E}(\mathcal{S}_T)$.*

Theorem 8.3 *For any recursive NRS system $\mathcal{S} = \langle U, N \rangle$, there is a recursive tree $T_{\mathcal{S}}$ such that there is an effective 1:1 degree preserving correspondence between $[T_{\mathcal{S}}]$ and $\mathcal{E}(\mathcal{S})$. Vice versa, for any recursive tree T, there is a recursive NRS system $\mathcal{S}_T = \langle U, N \rangle$ such that there is an effective 1:1 degree preserving correspondence between $[T]$ and $\mathcal{E}(\mathcal{S}_T)$.*

Because the set of degrees of paths through recursive trees have been extensively studied in the literature, we immediately can derive a number of corollaries about the degrees of extensions in recursive NRS systems. We shall give a few of these corollaries below.

For recursive nonmonotonic rule systems, we have the following results, see [MNR92b].

Corollary 8.4 *1. Every recursive NRS system $\mathcal{S} = \langle U, N \rangle$ which has an extension has an extension E such that $E \leq_T B$ where B is a complete Π^1_1-set.*

2. If $\mathcal{S} = \langle U, N \rangle$ is a recursive NRS system with a unique extension E, then E is hyperarithmetic.

Corollary 8.5 *1. There is a recursive NRS system $\mathcal{S} = \langle U, N \rangle$ such that $\mathcal{S}$ has an extension but $\mathcal{S}$ has no extension which is hyperarithmetic.*

2. For each recursive ordinal α, there exists a recursive NRS system $\mathcal{S} = \langle U, N \rangle$ possessing a unique extension E such that $E \equiv_T \mathbf{0}^{(\alpha)}$.

These two corollaries show that the extensions of a recursive nonmonotonic rule system may be very complex. There are natural conditions which will guarantee that the set of extensions of a nonmonotonic rules system are much better behaved. For example, if the nonmonotonic rule system is highly recursive, then we have the following results, see [MNR92b].

Call A *low* if $A' \equiv_T \mathbf{0}'$. The following corollary is an immediate consequence of Theorem 8.1 and the work of Jockusch and Soare [JS72b].

Corollary 8.6 *Let $\mathcal{S} = \langle U, N \rangle$ be a highly recursive nonmonotonic rule system such that $\mathcal{E}(\mathcal{S}) \neq \emptyset$. Then*
(i) There exists an extension E of $\mathcal{S}$ such that E is low.
(ii) If $\mathcal{S}$ has only finitely many extensions, then every extension E of $\mathcal{S}$ is recursive.

In the other directions, there are a number of corollaries of the Theorem 8.1 which allow us to show that there are highly recursive NRS systems $\mathcal{S}$ such that the set of degrees realized by elements of $\mathcal{E}(\mathcal{S})$ are still quite complex. Again all these corollaries follow by transferring results of Jockusch and Soare [JS72b, JS72a].

Corollary 8.7 *1. There is a highly recursive nonmonotonic rule system $\langle U, N\rangle$ such that $\langle U, N\rangle$ has $2^{\aleph_0}$ extensions but no recursive extensions.*

2. *There is a highly recursive nonmonotonic rule system $\langle U, N\rangle$ such that $\langle U, N\rangle$ has $2^{\aleph_0}$ extensions and any two extensions $E_1 \neq E_2$ of $\langle U, N\rangle$ are Turing incomparable.*

3. *There is a highly recursive nonmonotonic rule system $\langle U, N\rangle$ such that $\langle U, N\rangle$ has $2^{\aleph_0}$ extensions and if $\mathbf{a}$ is the degree of any extension E of $\langle U, N\rangle$ and $\mathbf{b}$ is any recursively enumerable degree such that $\mathbf{a} <_T \mathbf{b}$, then $\mathbf{b} \equiv_T \mathbf{0}'$.*

4. *If $\mathbf{a}$ is any recursively enumerable Turing degree, then there is a highly recursive nonmonotonic rule system $\langle U, N\rangle$ such that $\langle U, N\rangle$ has $2^{\aleph_0}$ extensions and the set of recursively enumerable degrees $\mathbf{b}$ which contain an extension of $\langle U, N\rangle$ is precisely the set of all recursively enumerable degrees $\mathbf{b} \geq_T \mathbf{a}$.*

Finally, we note that there are analogues of Corollaries 8.6 and 8.7 which hold for recursive locally finite nonmonotonic rule systems. That is, one can replace highly recursive nonmonotonic rule systems by recursive locally finite nonmonotonic rule systems if one replaces all the statements about degrees of extensions by the corresponding statement relative to an $\mathbf{0}'$ oracle. For example, the analogue of part (1) of Corollary 8.6 is that every recursive locally finite nonmonotonic rule system $\mathcal{S}$ such that $\mathcal{E}(\mathcal{S}) \neq \emptyset$ has an extension E such that the jump of E is recursive in $\mathbf{0}''$, while the analogue of (1) of Corollary 8.7 is that there exists a recursive locally finite nonmonotonic rule system $\langle U, N\rangle$ which has $2^{\aleph_0}$ extensions but which has no extension which is recursive in $\mathbf{0}'$. Moreover, we can weaken the hypothesis of locally finite and highly recursive slightly and still derive the same theorems. That is, we say that a recursive nonmonotonic rule system $\langle U, N\rangle$ has the *finite support property* if for each $c \in U$, the set of supports of all $\prec$-minimal proofs schemes of c is finite. It is possible for a $c \in U$ to have infinitely many $\prec$-minimal proof schemes with the same support so that not every recursive nonmonotonic rule system with the finite support property is locally finite. Similarly, we say that a recursive nonmonotonic rule system $\langle U, N\rangle$ which has the finite support property has the *recursive finite support property* if there is an effective algorithm which given any $c \in U$, produces the canonical index of the set of canonical indices of the supports of all the $\prec$-minimal proof schemes of c. See [MNR94b] for further details.

9 Extended recursive nonmonotonic rule systems and their recursion theory

In this section we investigate extended nonmonotonic rule systems $\langle U, N\rangle$ such that $U \subseteq \omega$ and each rule $r \in N$ has a recursive set of restraints. Specifically, we shall assume that all rules $r \in N$ are of the form

$$r = \frac{a_1, \ldots, a_n : R}{c} \tag{6}$$

where R is a set which may be finite or infinite recursive. We do not exclude the possibility that R is empty; in this case r is a monotonic rule. With these assumptions, we can define natural analogues for extended nonmonotonic rule systems of recursive, locally finite, and highly recursive nonmonotonic rule systems. The key to our definitions is a careful definition of the code of a rule and the code of a proof scheme.

Definition 9.1 Let $\mathcal{S} = \langle U, N\rangle$ be an extended nonmonotonic rule system such that $U \subseteq \omega$ and N is a set of rules r of the form (6) where $a_1, \ldots, a_n, c \in U$ and R is a recursive subset of U. A *code* $c(r)$ for a rule of the form (6) is the code of the quadruple $c(r) = [k, e, h, c]$ such that:

(i) $\{a_1, \ldots, a_n\} = D_k$,

(ii) $\varphi_e = \chi_R$ where φ_e is the e^{th} partial recursive function and χ_R is the characteristic function of R,

(iii) h is a natural number satisfying the following conditions:

1. If R is finite and nonempty, then $h = 2 + \max(R)$
2. If R is infinite, then $h = 0$
3. If R is empty, then $h = 1$

(iv) c is the conclusion of the rule.

Of course each rule possesses infinitely many codes $c(r)$ since there is infinitely many indices e such that $\varphi_e = \chi_R$. The code $c(r)$ contains the information whether the restraint set R is finite or infinite. Moreover if R is finite, then from the code $c(r)$ we can effectively compute a canonical index of R by computing the sequence $\varphi_e(0), \ldots, \varphi_e(h-2)$. Thus in the case of rules r with finite restraints, the codes carry all the information necessary to compute the rule r.

We shall now define the class of recursive extended nonmonotonic rule systems.

Definition 9.2 Let $\mathcal{S} = \langle U, N \rangle$ be an extended nonmonotonic rule system such that $U \subseteq \omega$ and N is a set of rules r of the form (6) where $a_1, \ldots, a_n, c \in U$ and R is a recursive subset of U. We say that the pair $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ is a **recursive extended nonmonotonic rule system** if

1. U is a recursive subset of ω
2. $\mathcal{C}(N)$ is a recursive set such that
   (a) $c \in \mathcal{C}(N)$ implies that c is a code of some rule in N
   (b) $r \in N$ implies that there is precisely one $c \in \mathcal{C}(N)$ such that c is a code for r

Thus our definition of recursive extended nonmonotonic rule system implies that there is a one-to-one correspondence between N and $\mathcal{C}(N)$. This correspondence takes a rule to its unique code in $\mathcal{C}(N)$ and allows us to identify the rule r as in (6) with its code $c(r)$ in $\mathcal{C}(N)$.

Recall that the definition of a proof scheme for an extended nonmonotonic rule system is the same as the definition of a proof scheme for a nonmonotonic rule system, see definition 6.1.

Definition 9.3 Let $\langle \mathcal{S}, \mathcal{C}(N)\rangle$ be a recursive extended nonmonotonic rule system where $\mathcal{S} = \langle U, N\rangle$. Let

$$p = \langle\langle c_0, r_0, Z_0\rangle, \ldots \langle c_m, r_m, Z_m\rangle\rangle$$

be a proof scheme for $c = c_n$ in $\mathcal{S}$. Then the code $c(p)$ is equal to

$$[m, [[c_0, c(r_0), e_0, n_0], \ldots, [c_m, c(r_m), e_m, n_m]]]$$

where

1. if $m = 0$, then either
   (a) $r_0 = \frac{:}{c_0}$ is an axiom and $c(r_0) = [0, e_0, n_0, c_0] \in \mathcal{C}(N)$ so that $n_0 = 1$ and $\varphi_{e_0} = \chi_\emptyset = \chi_{Z_0}$, or
   (b) $r_0 = \frac{:R}{c_0}$ where R is a nonempty recursive set and $c(r_0) = [0, e_0, n_0, c_0] \in \mathcal{C}(N)$ so that $\varphi_{e_0} = \chi_R = \chi_{Z_0}$ and $n_0 = 0$ if R is infinite and $n_0 = 2 + max(R)$ in R is finite,
2. if $m > 0$, then assume that

   $$[m-1, [[c_0, c(r_0), e_0, n_0], \ldots, [c_{m-1}, c(r_{m-1}), e_{m-1}, n_{m-1}]]]$$

   is the code of the proof scheme

   $$p = \langle\langle c_0, r_0, Z_0\rangle, \ldots \langle c_{m-1}, r_{m-1}, Z_{m-1}\rangle\rangle$$

   and

   $$\varphi_{e_{m-1}} = \chi_{Z_{m-1}}$$

   for the recursive set Z_{m-1} and

   $$n_{m-1} = \begin{cases} 0 & \text{if } Z_{m-1} \text{ is infinite} \\ 1 & \text{if } Z_{m-1} \text{ is empty} \\ 2 + \max(Z_{m-1}) & \text{otherwise} \end{cases}$$

   Then if

   $$r_m = \frac{c_{i_0}, \ldots, c_{i_t} : R_m}{c_m}$$

where $i_0 \ldots, i_t < m$ and $c(r_m) = [k_m, d_m, n_m, c_m] \in \mathcal{C}(N)$, then we let $e_m = f(e_{m-1}, d_m)$ where f is a total recursive function such that for all k_1 and k_2, $\varphi_{f(k_1,k_2)} = sg(\varphi_{k_1} + \varphi_{k_2})(x)$. Here we use Kleene's sg function which is defined by

$$sg(x) = \begin{cases} 1 & \text{if } x \geq 1 \\ 0 & \text{otherwise.} \end{cases}$$

Finally we let

$$n_m = \begin{cases} 0 & \text{if either } n_{m-1} \text{ or } n \text{ is } 0 \\ 1 & \text{if both } n_{m-1} \text{ and } n \text{ are equal to } 1 \\ \max(n_{m-1}, n) & \text{otherwise.} \end{cases}$$

Let us see what is the meaning of this definition. First notice the following property of the function f:

$$\varphi_{f(k_1,k_2)}(x) = \begin{cases} 1 & \text{if } \varphi_{k_i}(x) \downarrow \text{ for } i = 1,2 \text{ and } \varphi_{k_1}(x) + \varphi_{k_2}(x) \geq 1 \\ 0 & \text{if } \varphi_{k_i}(x) \downarrow \text{ for } i = 1,2 \text{ and } \varphi_{k_1}(x) + \varphi_{k_2}(x) = 0 \\ \text{undefined} & \varphi_{k_1}(x) \uparrow \text{ or } \varphi_{k_2}(x) \uparrow \end{cases}$$

This implies that

$$\varphi_{f(e_{m-1},d_m)} = \chi_{Z_{m-1} \cup R_m} = \chi_{Z_m}.$$

This, in turn, implies that the last index, e_m is an index of the characteristic function of the union of the restraints of all the rules applied in the proof scheme under consideration. Moreover our definition of n_m implies that:

$$n_m = \begin{cases} 0 & \text{if } Z_m = Z_{m-1} \cup R_m \text{ is infinite} \\ 1 & \text{if } Z_m = Z_{m-1} \cup R_m \text{ is empty} \\ 2 + \max(Z_m) & \text{if both } Z_{m-1},\ R_m \text{ are finite} \end{cases}$$

It is easy to check that we can effectively decide if p is an extended proof scheme. That is, given a number z, we can decode from z numbers s and u such that $z = [s, u]$, then we can decode from u a sequence (of the length $s + 1$) of quadruples,

$$u = \langle\langle c_0, a_0, e_0, n_0\rangle, \ldots, \langle c_s, a_s, e_s, n_s\rangle\rangle$$

and check if $c_i \in U$ and $a_i \in \mathcal{C}(N)$ for all $i \leq s$, and then compute to see that c_i, e_i, and n_i satisfy the required conditions.

Moreover, we can define a preorder $\prec$ on proof schemes in extended nonmonotonic rule system just as we did in section 6. This, in turn, leads us to the definition of locally finite and highly recursive extended nonmonotonic systems.

Definition 9.4 Let $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ be a recursive extended nonmonotonic rule system where $\mathcal{S} = \langle U, N \rangle$.

1. We say that $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ is *locally finite* if for every $a \in U$ there are finitely many $\prec$-minimal proof schemes in $\mathcal{S}$ with conclusion a. We let Dr_a denote the set of all codes of $\prec$-minimal proof schemes for a in $\mathcal{S}$.
2. We say that $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ is a *highly recursive* ENRS if $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ is a locally finite ENRS and the map $a \mapsto can(Dr_a)$ is partial recursive.

We are now in a position to prove the main results of our paper. Our first result will show that the expressive power of highly recursive extended nonmonotonic rule systems is at least as great as that of recursive nonmonotonic rule systems. This is in stark contrast with the results of section 6 where it is shown that the expressive power of recursive nonmonotonic rule systems is much greater than the expressive power of highly recursive nonmonotonic rule systems.

Theorem 9.5 *Let T be a recursive tree, $T \subseteq \omega^{<\omega}$. Then there is a highly recursive ENRS $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ where $\mathcal{S} = \langle U, N \rangle$ such that there exists an effective, one-to-one, degree-preserving correspondence between $\mathcal{E}(\mathcal{S})$ and the set $[T]$.*

Proof: We shall assume that all trees $T \subseteq \omega^{<\omega}$ contain the empty sequence, $\emptyset$. First we shall modify the tree T as follows. Given any node $\sigma = (\sigma_0, \ldots, \sigma_k) \neq \emptyset$, we let $shift(\sigma) = (\sigma_0 + 1, \ldots, \sigma_k + 1)$. We let $T' = \{\emptyset\} \cup \{shift(\sigma) : \sigma \in T \setminus \{\emptyset\}\}$. We observe that T' is a recursive tree if and only if T is a recursive tree. Moreover, it is clear that there is a recursive, one-to-one, degree preserving correspondence between $[T]$ and $[T']$.

Next, we define $T'' = T' \cup \{\eta^\frown 0 : \eta \in T'\}$ where for any node $\eta \in \omega^{<\omega}$, $\eta^\frown a$ denotes the result of concatenating a at the end of η. Clearly T'' is also a recursive tree. Our construction ensures that all the coordinates of all the sequences in T' are greater or equal to 1. In T'', every node σ will have at least one successor node, unless σ ends with 0 in which case σ is a terminal node, i.e. σ has no successor. This implies that in T'' the set of nodes with at least one successor is recursive. On the other hand it is easy to see that $[T'] = [T'']$, so that there is a one-to-one, degree-preserving correspondence between $[T]$ and $[T'']$.

We are ready to construct the desired extended nonmonotonic rule system $\mathcal{S}$. First, we define U. $U'' = \{\emptyset\} \cup \{\eta : \eta \in T'' \setminus \{\emptyset\}\} \cup \{\bar{\eta} : \eta \in T'' \setminus \{\emptyset\}\}$. Notice that for every sequence $\eta \in T'' \setminus \{\emptyset\}$ we have in U'' *two* different objects: η itself, and $\bar{\eta}$. This "duplicate" $\bar{\eta}$ is used as an indicator that η does not belong to the putative path (i.e. extension). Then U equals the set of all codes $c(\sigma)$ of elements $\sigma \in U''$ where

1. $c(\emptyset) = 0$
2. if $\sigma = \langle \sigma_0, \ldots, \sigma_k \rangle$ then $c(\sigma) = [0, k, [\sigma_0, \ldots, \sigma_k]]$
3. if $\sigma = \langle \sigma_0, \ldots, \sigma_k \rangle$ then $c(\bar{\sigma}) = [1, k, [\sigma_0, \ldots, \sigma_k]]$.

It is easy to see that U is a recursive set since T'' is a recursive tree. To simplify notation, for the rest of this proof we shall implicitly identify U'' and U by identifying each element of U'' with its code. Similarly, when we talk of proof schemes, we will implicitly identify a proof scheme and its code.

Now we define N. N consists of the following 7 classes of rules

1. $\dfrac{:}{\emptyset}$
2. $\dfrac{:}{\eta^\frown 0}$ for any $\eta \in T''$ such that η possesses at least one successor in T''.
3. $\dfrac{\eta : \{\eta^\frown i : \eta^\frown i \in T'' \text{ and } i \neq j\}}{\eta^\frown j}$ for all $\eta \in T''$ and $j \in \omega$ such that η has at least one successor in T'', $\eta^\frown j \in T''$, and $j > 0$.

4. $\dfrac{\eta : \{\eta^\frown i : \eta^\frown i \in T''\}}{\eta^\frown 0}$ for all $\eta \in T''$ such η has at least one successor in T''.

5. $\dfrac{: \eta}{\bar{\eta}}$ and $\dfrac{: \bar{\eta}}{\eta}$ for all $\eta \in T'' \setminus \{\emptyset\}$.

6. $\dfrac{\eta^\frown i, \eta^\frown j :}{\overline{\eta^\frown j}}$ for all $\eta^\frown i, \eta^\frown j \in T''$ with $i < j$.

7. $\dfrac{\bar{\eta} :}{\overline{\eta^\frown j}}$ for all $\eta, \eta^\frown j \in T''$ such that $|\eta| \geq 1$.

It is easy to see that since T'' is recursive tree and $\{\sigma \in T'' : \sigma$ has at least one successor in $T''\}$ is also recursive (see our remarks on the set of terminal nodes in T''), one can easily use any index of the characteristic function of T'' to produce a recursive set of codes $\mathcal{C}(N)$ which are in one to one correspondence with the rules of N. Therefore $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ is a recursive extended nonmonotonic rule system.

Next we shall prove that $\langle \mathcal{S}, \mathcal{C}(N) \rangle$ is highly recursive. We must show that each element of U is the conclusion of only finitely many minimal proof schemes and the we can effectively compute that set of minimal proof schemes. It should be clear that, as $\emptyset$ is a conclusion of just one rule, $Dr_\emptyset$ is finite. In fact, it has exactly one element corresponding to the proof scheme $\langle \emptyset, \frac{:}{\emptyset}, \emptyset \rangle$. Next we describe how to compute Dr_σ and $Dr_{\overline{\sigma}}$ for all $\sigma \in T'' \setminus \{\emptyset\}$.

Fix η and assume that Dr_η is finite and that we have effectively computed $Can(Dr_\eta)$ from η. Consider a successor node, $\eta^\frown j \in T''$. First consider the case when $j = 0$. Note that $\eta^\frown 0 \in T''$. It should be clear that the nodes of the form $\eta^\frown 0$ can be derived only using the rules of the form (4) or (5). Moreover the rules of the form (5) have no premises. This means that a minimal proof scheme for $\eta^\frown 0$ whose last rule is of the form (5) must be of the form

$$\langle \langle \eta^\frown 0, \frac{: \overline{\eta^\frown 0}}{\eta^\frown 0}, \{\overline{\eta^\frown 0}\} \rangle \rangle.$$

If the last rule in a minimal proof scheme of $\eta^\frown 0$ is a rule of the form of (4), then the proof scheme must be of the form

$$\langle \langle a_0, r_0, Z_0 \rangle, \ldots, \langle a_m, r_m, Z_m \rangle \rangle$$

where $a_m = \eta^\frown 0$ and r_m of type (4). But then

$$\langle\langle a_0, r_0, Z_0\rangle, \ldots, \langle a_{m_1}, r_{m-1}, Z_{m-1}\rangle\rangle$$

is a minimal proof scheme for η. Since Dr_η is finite, this implies that $Dr_{\eta^\frown 0}$ is finite. Moreover, clearly we can effectively compute $Dr_{\eta^\frown 0}$ from Dr_η.

If $j \geq 1$ then we can derive $\eta^\frown j$ only using the rules of form (3) or (5). If the last rule in minimal proof scheme of $\eta^\frown j$ is a rule of the form of (5), then we can argue exactly as above that the minimal proof scheme must be of the form

$$\langle\langle \eta^\frown j, \frac{:\overline{\eta^\frown j}}{\eta^\frown j}, \{\overline{\eta^\frown j}\}\rangle\rangle.$$

So there is just one minimal proof scheme of $\eta^\frown j$ whose last rule is of type (5).

If the last rule in minimal proof scheme of $\eta^\frown j$ is a rule of the form of (3), then the minimal proof scheme must be of the form

$$\langle\langle a_0, r_0, Z_0\rangle, \ldots, \langle a_m, r_m, Z_m\rangle\rangle$$

where $a_m = \eta^\frown j$ and r_m of type (3). But then

$$\langle\langle a_0, r_0, Z_0\rangle, \ldots, \langle a_{m_1}, r_{m-1}, Z_{m-1}\rangle\rangle$$

must be a minimal proof scheme for η. Thus the set of minimal proof schemes that use the rule (3) is also finite and can be effectively computed once Dr_η is computed. Hence we can conclude that $Dr_{\eta^\frown j}$ is finite and that we have an effective method of computing $Dr_{\eta^\frown j}$.

Next, consider the elements of the form $\bar{\alpha}$. By construction, $\alpha = \eta^\frown j$ for some j. Notice that only rules of the form (2), (5), (6) and (7) can be used to derive $\bar{\alpha}$.

We need to look at each of these cases.

First suppose the last rule used in a minimal proof scheme of $\overline{\alpha}$ is a rule of the form (2). There is only one rule of the form (2) with the conclusion $\bar{\alpha}$. Moreover since all rules in (2) have no premises, it follows that there is just one minimal proof scheme with conclusion $\bar{\alpha}$ whose last rule is of type (2).

If the last rule used in a minimal proof scheme of $\overline{\alpha}$ is a rule of the form (5), then we can reason exactly as in the case when we used the rule (5) to get α to conclude that there is only one such proof scheme.

Suppose the last rule used in a minimal proof scheme of $\overline{\alpha}$ is a rule of the form (7), then it is easy to that the minimal proof scheme must consist of a minimal proof scheme for $\bar{\eta}$ extended by a single triple of the form $< \bar{\alpha}, r_m, Z_m >$ where $r_m = \frac{\bar{\eta}:}{\bar{\alpha}}$.

Finally consider the case when the last rule of the proof scheme is a rule of the form (6). Here we use the inductive assumption. There is only $j+1$ sets $Dr_{\eta^\frown i}$ with $i \leq j$ which are involved in the computation of of premises of the rule of the form (6) with the conclusion $\bar{\alpha}$. Each of these sets is finite. Every minimal proof scheme for $\overline{\eta^\frown j}$ using the rule of the type (6) with i, j arises from interveawing a minimal proof scheme for $\eta^\frown i$ and a minimal proof scheme for $\eta^\frown j$ and putting a corresponding rule of the type (6) at the end (and adjusting the bookkeeping). There may be, additionally, some pruning necessary if the same rules were used in both proof schemes. Thus it is clear that there are only finitely many minimal proof schemes for $\bar{\alpha}$ in which the last rule used of the form (6). Moreover, our analysis provides an effective method of finding this set of minimal proof schemes from the set of minimal proof schemes for $\eta\bar{^\frown} i$ for $i \leq j$.

Putting together all these cases, it follows that the set $Dr_{\bar{\alpha}}$ is finite and that there is an effective method to compute it. Thus, we conclude that $\mathcal{S}$ is a highly recursive, extended nonmonotonic rule system.

Next, we shall show that there exists an effective, one-to-one correspondence between $\mathcal{E}(\langle U, N\rangle)$ and $[T'']$. In particular we will show that $\mathcal{E}(\langle U, N\rangle) = \emptyset$ if and only if $[T''] = \emptyset$. First, suppose that $\pi = (\pi_0, \pi_1, \ldots)$ is an infinite path through T''. Define $\eta_i = (\pi_0, \pi_1, \ldots, \pi_i)$. Then we claim that

$$E_\pi = \{\emptyset\} \cup \{\eta_i : i \in \omega\} \cup \{\bar{\eta} : \eta \in T'' \setminus (\{\emptyset\} \cup \{\eta_i : i \in \omega\})\}$$

is an extension of $\langle U, N\rangle$. To see that all elements of E_π have an E_π-derivation in $\langle U, N\rangle$, note that $\emptyset \in C_{E_\pi}(\emptyset)$ by rule (1). Next, it is easy to see by induction that the rules of the form (3) allow us to prove that $\eta_i \in C_{E_\pi}(\emptyset)$ for all $i \in \omega$. Finally, the rules from the group (5) will allow us to prove that $\bar{\eta} \in C_{E_\pi}(\emptyset)$ for all $\eta \in T'' \setminus (\{\emptyset\} \cup \{\eta_i : i \in \omega\})$. Thus $E_\pi \subseteq C_{E_\pi}(\emptyset)$.

To see that $C_{E_\pi}(\emptyset) \subseteq E_\pi$, we proceed by induction on the length of a sequence. That is, assume by induction that

1. $\eta \in C_{E_\pi}(\emptyset)$ where $\eta \in T''$ and $|\eta| \leq k$ if and only if $\eta \in \{\emptyset, \eta_0, \ldots, \eta_{k-1}\}$ and
2. $\bar{\eta} \in C_{E_\pi}(\emptyset)$ where $\eta \in T'' \setminus \{\emptyset\}$ and $|\eta| \leq k$ if and only if $\eta \in T'' \setminus \{\emptyset, \eta_0, \ldots, \eta_{k-1}\}$.

Next observe that if $|\alpha| = k+1$, then the only rules which have α as the conclusion are the rules of the form (3), (4), and (5). Since $|\alpha| = k+1$, there is β and $j \in \omega$ such that $\alpha = \beta^\frown j$.

Let us look at the rules in question. These are:

(a) $\dfrac{\beta : \{\beta^\frown i : \beta^\frown i \in T'' \textit{ and } i \neq j\}}{\beta^\frown j}$ from the group (3), if β has at least one successor in T'' and $j > 0$.

(b) $\dfrac{\beta : \{\beta^\frown i : \beta^\frown i \in T''\}}{\beta^\frown 0}$ from the group (4), if β has at least one successor in T'' and $j = 0$

(c) $\dfrac{: \overline{\beta^\frown j}}{\beta^\frown j}$ from the group (5).

Now, if $\beta \neq \eta_{k-1}$ then the rules (a) and (b) can not be used to show $\beta^\frown j \in C_{E_\pi}(\emptyset)$ since by assumption $\beta \notin C_{E_\pi}(\emptyset)$. Also the rule (c) is blocked since $\overline{\beta^\frown j} \in E_\pi$. Thus if $\beta \neq \eta_{k-1}$ then $\beta^\frown j \notin C_{E_\pi}(\emptyset)$. If $\beta = \eta_{k-1}$, but $j \neq \pi_k$, then the rules (a) and (b) are blocked by E_π since $\eta_k = \eta_{k-1}{}^\frown\pi_k \in E_\pi$. Similarly the rule (c) is blocked for $j \neq \pi_k$ because in this case $\overline{\beta^\frown j} \in E_\pi$. This shows that $\beta^\frown j \in C_{E_\pi}(\emptyset)$ if and only if $\beta^\frown j = \eta_k$. Note that for all α with $|\alpha| = k+1$ and $\alpha \neq \eta_k$, we have that $\alpha \notin E_\pi$. Using the rules of (5), we can derive that $\bar{\alpha} \in C_{E_\pi}(\emptyset)$ for such α.

Finally we must check that $\overline{\eta_k} \notin C_{E_\pi}(\emptyset)$. But the only rules that have $\overline{\eta_k}$ as the conclusion are the rules form (5), (6), and (7), namely

(A) $\dfrac{: \eta_k}{\overline{\eta_k}}$ from (5),

(B) $\dfrac{\eta_{k-1}^{\frown}i, \eta_{k-1}^{\frown}\pi_k :}{\overline{\eta_{k-1}^{\frown}\pi_k}}$ from (6), where $i < \pi_k$ (recall that $\eta_k = \eta_{k-1}^{\frown}\pi_k$), and

(C) $\dfrac{\overline{\eta_{k-1}} :}{\overline{\eta_k}}$ from the group (7).

(Note that the rule of the form (2) cannot be used because $\pi_k > 0$. This last fact follows since η_k possesses a successor).

The rule (A) is blocked for E_π since $\eta_k \in E_\pi$. Rule (B) cannot be used because we have already shown $\eta_{k-1}^{\frown}i \notin C_{E_\pi}(\emptyset)$ for every $i \neq \pi_k$. Finally, rule (C) cannot be used to show $\overline{\eta_k} \in C_{E_\pi}(\emptyset)$ because by our induction hypothesis, $\overline{\eta_{k-1}} \notin C_{E_\pi}(\emptyset)$.

Thus we have shown that for all $\alpha \in T''$ such that $|\alpha| = k+1$, $\alpha \in C_{E_\pi}(\emptyset)$ if and only if $\alpha = \eta_k$ and $\overline{\alpha} \in C_{E_\pi}(\emptyset)$ if and only if $\alpha \neq \eta_k$. Therefore, by induction, $C_{E_\pi}(\emptyset) \subseteq E_\pi$ and hence we can conclude that E_π is an extension of $\langle U, N\rangle$.

Next, suppose that E is an extension of $\langle U, N\rangle$. We claim that E must be of the form E_π where π is some infinite path through T''. We shall prove by induction on k that for each k there is a unique node δ_k of the length k such that $\delta_k \in E$ and that all these sequences δ_k are compatible. Moreover, we shall show that if $k > 0$ and $|\beta| = k$, then $\bar{\beta} \in E$ if and only if $\beta \neq \delta_k$. Note that $\emptyset$ is a unique node of the length 0 and $\emptyset \in E$ by the rule (1).

So assume that there is exactly one node δ_k such that δ_k is of length k, $\delta_k \in E$, and δ_k has at least one successor and that $|\beta| = k$ and $\beta \neq \delta_k$ implies $\bar{\beta} \in E$. Now suppose that $|\alpha| = k + 1$ and $\alpha = \beta^\frown j$.

First, suppose $\beta \neq \delta_k$. Then, by our analysis, $\bar{\beta} \in E$ and hence the rules of the form (7) allow us to prove that for all j such that $\beta^\frown j \in T''$, $\overline{\beta^\frown j} \in E$. Now consider the rules (a), (b), (c) with conclusion $\beta^\frown j$. Note that the rules of the form (a), and (b) can not be used to put $\beta^\frown j$ into $E = C_E(\emptyset)$ since $\beta \notin E = C_E(\emptyset)$ by our assumption. Now by rule (7), $\overline{\beta^\frown j} \in E$ since $\overline{\beta} \in E$ and hence the rule (c) is blocked for E. Thus $\beta^\frown j \notin C_E(\emptyset) = E$.

Next, assume $\beta = \delta_k$. We claim that it cannot be the case that $E \cap \{\delta_k{}^\frown j : \delta_k{}^\frown j \in T''\} = \emptyset$. Indeed, otherwise by rule (4) for $\eta = \delta_k$, we must conclude that $\delta_k{}^\frown 0 \in C_E(\emptyset)$. But since δ_k has at least one successor, $\delta_k^\frown 0 \in T''$ and hence $\delta_k{}^\frown 0 \in \{\delta_k{}^\frown j : \delta_k{}^\frown j \in T''\}$. But then $C_E(\emptyset) \neq E$ which would contradict the fact that E is an extension. Next define $F_k = E \cap \{\delta_k{}^\frown j : \delta_k{}^\frown j \in T''\}$. We just proved that $F_k \neq \emptyset$. Suppose that $\delta_k{}^\frown i, \delta_k{}^\frown j \in F_k$ where $i < j$. Then by rule (6), the fact that $\delta_k{}^\frown i, \delta_k{}^\frown j \in E = C_E(\emptyset)$ implies that $\overline{\delta_k{}^\frown j} \in C_E(\emptyset) = E$. However if we consider the possible of the form (a), (b), (c) where $\beta = \delta_k$ which could have $\delta_k{}^\frown j$ as a conclusion, we see that they are all blocked by E. But then we would have to conclude $\delta_k{}^\frown j \notin C_E(\emptyset)$ which again contradicts the fact that E is an extension of $\langle U, N \rangle$. Hence we must conclude that there is a unique j such that $\delta_k{}^\frown j \in E$. Of course, we can now use the rules of the form (5) to conclude that if $i \neq j$ and $\delta_k{}^\frown i \in T''$ then $\overline{\delta_k{}^\frown i} \in C_E(\emptyset) = E$. At this stage of our argument we have shown that there is a unique node of the length $k+1$, $\delta_k{}^\frown j$ such that $\delta_k{}^\frown j \in E$. Moreover, for all nodes α of the length $k+1$ $\alpha \neq \delta_k{}^\frown j$ implies $\bar{\alpha} \in E$. Thus to complete our induction step we need to prove two things. First, we must show that $j \neq 0$ so that $\delta_k{}^\frown j$ is guaranteed to have at least one successor. Second, we must show that $\overline{\delta_k{}^\frown j} \notin E$. There are exactly two rules which have $\delta_k{}^\frown 0$ as a conclusion:

(I) $\dfrac{\delta_k : \{\delta_k{}^\frown i : \delta_k{}^\frown i \in T''\}}{\delta_k{}^\frown 0}$ and

(II) $\dfrac{: \overline{\delta_k{}^\frown 0}}{\delta_k{}^\frown 0}$

But if we assume that $j = 0$ so that $\delta_k{}^\frown 0 \in E$ then the rule (I) is blocked for E. However since E is an extension, the rule (2) with $\eta = \delta_k$ implies $\overline{\delta_k{}^\frown 0} \in E$ and hence the rule (II) is also blocked for E. But this means that $\delta_k{}^\frown 0 \notin C_E(\emptyset)$ and hence $C_E(\emptyset) \neq E$. Thus $j > 0$.

Finally, to see that $\overline{\delta_k{}^\frown j} \notin E$, note that the only rules which have $\overline{\delta_k{}^\frown j}$ as a conclusion are:

(III) $\dfrac{: \delta_k{}^\frown j}{\overline{\delta_k{}^\frown j}}$ form the group (5)

(IV) $\dfrac{\delta_k{}^\frown i, \delta_k{}^\frown j :}{\overline{\delta_k{}^\frown j}}$ from the group (6) where $i < j$ and $\delta_k{}^\frown i \in T''$, or

(V) $\dfrac{\overline{\delta_k} :}{\overline{\delta_k \frown j}}$ from the group (7)

But now we can argue exactly as we did for $\overline{\delta_k \frown \pi_k}$, that none of the rules (III), (IV), or (V) can apply to show $\overline{\delta_k \frown j} \in C_E(\emptyset)$ given our assumptions. Hence $\overline{\delta_k \frown j} \notin C_E(\emptyset)$, as desired. This completes our induction and shows that $E = E_\Delta$ where Δ is an infinite path through T'' determined by the sequence $\emptyset \sqsubseteq \delta_1 \sqsubseteq \delta_2 \sqsubseteq \ldots$.

Thus we have established that the correspondence $\pi \mapsto E_\pi$ is a one-to-one correspondence between $[T]$ and $\mathcal{E}(\langle U, N\rangle)$. Clearly the correspondence is effective and degree-preserving. $\square$

Corollary 9.6 *There is a highly recursive extended nonmonotonic rule system $\langle\langle U, N\rangle, \mathcal{C}(N)\rangle$ such that $\langle U, N\rangle$ has an extension but $\langle U, N\rangle$ has no hyperarithmetic extension.*

Proof: Kleene (see [Rog67], Corollary XLI(b)), showed that there is a recursive tree $T \subseteq \omega^{<\omega}$ such that $[T] \neq \emptyset$ but no path through T is hyperarithmetical. Let $\langle\langle U, N\rangle, \mathcal{C}(N)\rangle$ be the extended nonmonotonic rule system constructed for T in the proof of Theorem 9.5. Then if $\langle U, N\rangle$ possesses a hyperarithmetic extension E, then there is an infinite path π through T such that $\pi \equiv_T E$. But the set of all hyperarithmetical subsets of ω is closed under Turing equivalence. Therefore π is itself hyperarithmetical, contradicting the choice of T. Thus $\langle U, N\rangle$ has no hyperarithmetic extension. $\square$

Corollary 9.7 *For every recursive ordinal α there is a highly recursive extended nonmonotonic rule system $\langle\langle U_\alpha, N_\alpha\rangle, \mathcal{C}(N_\alpha)\rangle$ such that $\langle U_\alpha, N_\alpha\rangle$ has a unique extension E_α and $E_\alpha \equiv_T 0^{(\alpha)}$ (where $0^{(\alpha)}$ is the α^{th} jump of of the recursive sets).*

Proof: Clote [Clo85] showed that for every recursive ordinal α, there exists a recursive tree T_α such that T_α has exactly one infinite path π_α and $\pi_\alpha \equiv_T 0^{(\alpha)}$. The corollary follows by letting $\langle\langle U_\alpha, N_\alpha\rangle, \mathcal{C}(N_\alpha)\rangle$ be the highly recursive extended nonmonotonic rule system constructed in the proof of Theorem 9.5 with $T = T_\alpha$. $\square$

Corollary 9.8 *The problem of deciding uniformly whether a highly recursive extended nonmonotonic rule system has an extension is Σ^1_1-complete, i.e. the set W of codes of highly recursive extended nonmonotonic rule systems which possess an extension is a Σ^1_1-complete set. (Here a number x is a code for a highly recursive system $\langle\langle U,N\rangle,\mathcal{C}(N)\rangle$ if $x =< u,n >$ where $\varphi_u = \chi_U$ and $\varphi_n = \chi_{\mathcal{C}(N)}$).*

Proof: It is easy to see that $W \in \Sigma^1_1$, by simply writing a definition. Note that our construction of Theorem 9.5 is uniform and hence there is a recursive function g such that if φ_e is a characteristic function of an infinite tree $T \subseteq \omega^{<\omega}$, then $g(e)$ is a code for a highly recursive extended nonmonotonic rule system constructed in the proof of Theorem 9.5.

Now [Rog67], Theorem XX, shows that $E = \{e : \varphi_e$ is a characteristic function of a recursive tree on ω such that $[T] = \emptyset\}$ is a complete Π^1_1 set. In fact, for each $A \in \Pi^1_1$ set Rogers constructs a recursive function h such that $\varphi_{h(e)}$ is always a characteristic function of a recursive tree on ω and $x \in A \Leftrightarrow h(x) \in E$. But this means that $x \notin A \Leftrightarrow h(x) \in P = \{x : \varphi_x$ is the characteristic function of a recursive tree $T \subseteq \omega^{<\omega}$ such that $[T] \neq \emptyset\}$. That is P is a Σ^1_1-complete set. By Theorem 9.5, $x \in P$ if and only if $g(x) \in W$. Thus W is Σ^1_1-complete. $\square$

Now we present a weak converse to Theorem 9.5 for recursive extended nonmonotonic rule systems.

Theorem 9.9 *Let $\langle\mathcal{S},\mathcal{C}(N)\rangle$ be a recursive extended nonmonotonic rule system where $\mathcal{S} = \langle U,N\rangle$. Then there is a recursive tree $T \subseteq \omega^{<\omega}$ and a pair of indices e and f such that*

1. *for any $\pi \in [T]$, $\varphi^\pi_e = \chi_{E_\pi}$ (where E_π is an extension of $\mathcal{S}$ and $E_\pi \leq_T \pi$, and*
2. *for any extension E of $\mathcal{S}$, $\varphi^{E'}_f = \chi_{\pi_e}$ where $\pi_E \in [T]$ and $\pi_E \leq_T E'$ (where E' is the jump of E).*

Moreover for all $E \in \mathcal{E}(\mathcal{S})$, $E_{\pi_E} = E$, and for all $\pi \in [T]$, $\pi_{E_\pi} = \pi$.

Before we provide the proof of this theorem, let us notice that all our theorem says is that $E \mapsto \pi_E$ is an effective, one-to-one correspondence between $\mathcal{E}(\mathcal{S})$ and $[T]$ but it only satisfies the inequality $E \leq_T \pi_E \leq_T E'$ rather than $\pi_E \equiv_T E$ as in the effective one-to-one degree preserving correspondence constructed in Theorem 9.5)

Proof of Theorem 9.9. There is no loss of generality in assuming that $U = \omega$. For if $U \neq \omega$, we simply consider the nonmonotonic rule system $\langle \omega, N \rangle$. Note that if $a \in \omega \setminus U$, then a is not a premise, restraint or conclusion of any rule in N. This implies that for every $c \in U$ the set of minimal proof schemes with conclusion c with respect to $\langle \omega, N \rangle$ is identical with the set of minimal proof schemes with conclusion c with respect to $\langle U, N \rangle$. Moreover, for $a \in \omega \setminus U$, the set of minimal proof schemes with respect to $\langle \omega, N \rangle$ with the conclusion a is empty. Hence it is easy to see that $\langle \langle \omega, N \rangle, \mathcal{C}(N) \rangle$ is (highly recursive) recursive extended nonmonotonic rule system if and only if $\langle \langle U, N \rangle, \mathcal{C}(N) \rangle$ is (highly recursive) recursive extended nonmonotonic rule system. Moreover, it follows that E is an extension of $\langle \omega, N \rangle$ if and only if E is an extension of $\langle U, N \rangle$. Thus we assume that $U = \omega$.

Notice that since $\langle \langle U, N \rangle, \mathcal{C}(N) \rangle$ is a recursive extended nonmonotonic rule system the set MP of codes all minimal proof schemes is a recursive set. In fact, for each $a \in U$, the set Dr_a of all codes of all minimal proof schemes p with $cln(p) = a$ is a recursive set. Moreover, we can find a recursive index for Dr_a uniformly form a. That is, there exists a total recursive function g such that $\varphi_{g(x)} = \chi_{Dr_a}$ for a $\in U$. Thus for any $a \in U$ and any code of a minimal proof scheme p with $cln(p) = a$, we can compute $\varphi_{g(a)}(0), \ldots, \varphi_{g(a)}(p)$ to find k such that $c(p)$ is the k^{th} element of Dr_a (i.e. $k = |\{x \leq c(p) : \varphi_{g(a)}(x) = 1\}|$), and produce minimal proof schemes $p_1, p_2, \ldots, p_{k-1}$ such that $c(p_1) < c(p_2) < \ldots < c(p_{k-1}) < c(p)$ and $c(p_i) \in Dr_a$ for $i = 1, \ldots, k-1$. Now suppose that E is an extension of $\langle U, N \rangle$ and $a \in E$. Then there is a minimal proof scheme q such that $cln(q) = a$ and $supp(q) \cap E = \emptyset$. Suppose that the q with the least code is p, the k^{th} element of Dr_a as described above. Then for each $i \leq k-1$ it must be the case that $supp(p_i) \cap E \neq \emptyset$. Now from $c(p_i)$, we can read off an index e_i such that $\varphi_{e_i} = \chi_{supp(p_i)}$, and hence we can effectively compute an index f_i such that $W^E_{f_i} = supp(p_i) \cap E$, for any E. From the jump of E, E', we can effectively decide if $W^E_{f_i} \neq \emptyset$ and if $W^E_{f_i} \neq \emptyset$, then we can use φ_{e_i} and E to find the least $x \in W^E_{f_i} = supp(p_i) \cap E$. To summarize, it follows that if E is an extension of a recursive extended nonmonotonic rule system $\langle\langle U, N\rangle, \mathcal{C}(N)\rangle$ and $a \in E$, then there is a uniform effective procedure which, given an oracle for E', will produce a pair $< p, z >$ such that

(i) $p = \mu x(x = c(q) \in Dr_a$ and $supp(q) \ \cap E \neq \emptyset)$ and

(ii) if p is the k^{th} element of Dr_a and $c(p_1) < \ldots < c(p_{k-1}) < c(q) = p$ are such that $c(p_i) \in Dr_a$ for $i = 1, \ldots, k-1$, then

 (a) $k = 1$ implies $z = 0$, and

 (b) $k > 1$ implies $z =< x_1, \ldots, x_{k-1} >$ where for each $1 \leq i \leq k-1$, $x_i = \mu y(y \in supp(p_i) \cap E)$.

Next, suppose that E is an extension of $\langle U, N\rangle$ and $a \notin E$. Then for any minimal proof scheme p such that $c(p) \in Dr_a$, it must be the case that $supp(p) \cap E \neq \emptyset$. Moreover, since from $c(p)$ we can read off an index e such that $\varphi_e = \chi_{supp(p)}$, we can effectively find the least element of the intersection $supp(p) \cap E$, given an oracle for E.

Our idea is to code an extension E by a path $\pi_E = (\pi_0, \pi_1, \ldots)$ through the complete ω-branching tree $T_\omega = \omega^{<\omega}$ as follows. First of all $\pi_{2a} = \chi_E(a)$ for all $a \in \omega$. Also, for all $a \in \omega$, $\pi_{2a+1} =< u_a, w_a >$ where if $a_1 < \ldots < a_n$ are all elements of $\{1, \ldots, a\} \setminus E$ then:

(I) $w_a =< x_1, \ldots, x_n >$ where for each i, $x_i =< b_{0,i}, \ldots, b_{a,i} >$ and for all $p \leq i$, $b_{p,i} = 0$ if $p \notin Dr_{a_i}$ and $b_{p,i} = 1 + \min(supp(q) \cap E)$ if q is a proof scheme for a_i such that $p = c(q) \in Dr_{a_i}$,

(II) if $\pi_{2a} = 1$ (so that $a \in E$), then $u_a =< p, z >$ where $< p, z >$ is the pair pair found from E' oracle for a as described above, and

(III) if $\pi_{2a} = 0$ (so that $a \notin E$), then $u_a = 0$.

Clearly $E \leq_T \pi_E$, and by our remarks above, $\pi_E \leq_T E'$. Thus the correspondence $E \mapsto \pi_E$ is a one-to-one correspondence which satisfies the requirements of the theorem. Hence to complete the proof of the theorem all we need to do is to produce a recursive tree $T^\star$ such that $[T^\star] = \{\pi_E : E \in \mathcal{E}(\langle U, N\rangle)\}$.

To define $T^\star$, first we ensure that $\emptyset \imath T^\star$. Then we put $\eta = \langle \eta_0, \ldots, \eta_k\rangle$ in $T^\star$ if and only if η statisfies the following conditions:

1. For all i such that $2i \leq k$, $\eta_{2i} \in \{0, 1\}$

2. For all i such that $2i+1 \leq k$, $\eta_{2i+1} =< u_i, w_i >$ where if $j_1 < \ldots < j_{n_i}$ is a list of all elements $j \leq i$ such that $\eta_{2j} = 0$, then $w_i =< x_1, \ldots, x_{n_i} >$ where for $s \leq n_i$, $x_s =< b_{0,j_s}, \ldots, b_{i,j_s} >$ where $b_{p,j_s} = 0$ if $p \notin Dr_{j_s}$ and $b_{p,j_s} - 1 \in supp(q)$ if q is a minimal proof scheme for j_s such that $p = c(q) \in Dr_{j_s}$. Moreover for each $t \leq i$, if $\eta_{2t+1} =< u_t, w_t >$ and $j_1 < \ldots < j_{n_t}$ is a list of all elements $j \leq t$ such that $\eta_{2j} = 0$, then $w_t =< y_1, \ldots, y_{n_t} >$ where for each $s \leq n_t$, $y_s =< b_{0,j_s}, \ldots, b_{t,j_s} >$. (This last requirement says that for a fixed j_s such that $\eta_{j_s} = 0$, the sequence associated to j_s, $(b_{0,j_s}, b_{1,j_s}, \ldots)$ is consistent).

3. For all i such that $2i + 1 \leq k$, if $\eta_{2i} = 0$ and $\eta_{2i+1} =< u_i, w_i >$ as described in (2), then $u_i = 0$

4. For all i such that $2i + 1 \leq k$, if $\eta_{2i} = 1$ and $\eta_{2i+1} =< u_i, w_u >$ as described in (2), then $u_i =< p, z >$ where $p = c(q)$ and q is a minimal proof scheme for i, $z = 0$ if p is the first element of Dr_i, and $z =< z_1, \ldots, z_{k-1} >$ if p is the k^{th} element of Dr_i for $k > 1$. Moreover, if $c(q_1) < c(q_2) < \ldots < c(q_{k-1}) < c(q) = p$ are all the elements of Dr_i smaller or equal than p, then for $i = 1, \ldots, k-1$, $z_i \in supp(q_i)$.

5. Let $E_\eta = \{i : 2i \leq k \wedge \eta_{2k} = 1\}$. Then for all i such that $2i + 1 \leq k$, if $\eta_{2i+1} =< u_i, w_i >$, $w_i =< x_1, \ldots, x_{n_i} >$ and for $s \leq n_i$ $x_s =< b_{0,j_s}, \ldots, b_{i,j_s} >$ are as described in (2), then for all $s \leq n_i$ and $p \leq i$ such that $b_{p,j_s} > 0$,

   (a) $p = c(q)$ for some minimal proof scheme of j_s and
   (b) $b_{p,j_s} - 1 > \lfloor \frac{k}{2} \rfloor$ and $E_\eta \cap supp(q) = \emptyset$ or
   (c) $b_{p,j_s} - 1 \leq \lfloor \frac{k}{2} \rfloor$ and b_{p,j_s} is the least element of $E_\eta \cap supp(q)$.

6. For all i such that $2i+1 \leq k$, if $\eta_{2i} = 1$, $\eta_{2i+1} =< u_i, w_i >$, $u_i =< p, z >$, q is the k^{th} element of Dr_i where $k > 1$, and $c(q_1) < \ldots < c(q_{k-1}) < c(q)$ and $z =< z_1, \ldots, z_{k-1} >$ are described in (4), then for all $t < k$

   (a) $z_t > \lfloor \frac{k}{2} \rfloor$ and $E_\eta \cap supp(q_t) = \emptyset$, or
   (b) $z_t \leq \lfloor \frac{k}{2} \rfloor$ and z_i is the least element of $supp(q_t) \cap E_\eta$.

   Moreover it must be the case that $E_\eta \cap supp(q) = \emptyset$.

It is routine to check that we can effectively decide if $\eta \in T^\star$ and that if $\eta \in T^\star$ and $\eta' \sqsubseteq \eta$, then $\eta' \in T^\star$. Thus $T^\star$ is a recursive tree. Moreover, it is easy to check that if E is an extension of $\langle U, N\rangle$ and $\pi_E = (\pi_0, \pi_1, \ldots)$, then $(\pi_0, \ldots, \pi_n) \in T^\star$ for all n. Thus

$$\{\pi_E : E \text{ is an extension of } \langle U, N\rangle\} \subseteq [T^\star]$$

Thus we only need to check that if $\pi = (\pi_0, \pi_1, \ldots)$ is an infinite path through $T^\star$, then $\pi = \pi_E$ for some extension E of $\langle U, N\rangle$.

So fix $\pi = (\pi_0, \pi_1, \ldots)$ and let $E = \{i : \pi_{2i} = 1\}$. Suppose that $i \notin E$ so that $\pi_{2i} = 0$. Assume that $Dr_i = \{p_0 = c(q_0), p_1 = c(q_1), \ldots\}$ is the listing of Dr_i in the increasing order where q_t is a minimal proof scheme for i for all t. Then for any r consider $\pi_{2n+1} =< u_n, w_n >$ where $n > \max(r, i)$. Now if i is the s^{th} element of $\omega \setminus E$ and $|\{i : 2i \leq n \wedge \pi_{2i} = 0\}| = t$, then $w_n =< x_1, \ldots, x_t >$ where $x_s =< b_{0,i}, \ldots, b_{t,i} >$ and $b_{r,i} - 1 \in supp(q_r)$. Moreover, if $\lfloor \frac{n}{2} \rfloor \geq b_{r,i} - 1$, then our conditions ensure that $b_{r,i} - 1$ is the least element of $E_{(\pi_0, \ldots, \pi_{2n+1})} \cap supp(q_r)$ which is the same as the least element of $E \cap supp(q_r)$. It follows that for each $p = c(q) \in Dr_i$, $E \cap supp(q) \neq \emptyset$ and hence $i \notin C_E(\emptyset)$. Thus we have shown that $i \notin E$ implies that $i \notin C_E(\emptyset)$. This means that $C_E(\emptyset) \subseteq E$.

Next, suppose that $i \in E$. Then $\pi_{2i} = 1$ and $\pi_{2i+1} =< u_i, w_i >$ where $u_i =< p, z >$ and $p = c(q) \in Dr_i$ for some minimal proof scheme of i. Moreover, if p is the k^{th} element of Dr_i, then $k = 1$ implies that $z = 0$ and $k > 1$ implies that $z =< z_1, \ldots, z_{k-1} >$ where $Dr_i = \{c(q_1) < \ldots < c(q_{k-1}) < c(q) < \ldots\}$ and $z_i \in supp(q_i)$, for $i = 1, \ldots, k-1$. The fact that $\delta_n = (\pi_0, \ldots, \pi_{2n+1}) \in T^\star$ for all $n \geq i$ implies that $E_{\delta_n} \cap supp(q) = \emptyset$ for all n by condition (6). Hence $E \cap supp(q) = \emptyset$, and q shows that $i \in C_E(\emptyset)$. Thus $E \subseteq C_E(\emptyset)$ and so the equality $E = C_E(\emptyset)$ holds. Thus E is an extension of $\langle U, N\rangle$. Moreover, if $k > 1$ and $m = \max\{z_1, \ldots, z_{k-1}\}$, then the fact that $\delta_m = (\pi_0, \ldots, \pi_{2m+1}) \in T^\star$ for all m implies that for $i < k$, z_i is the least element of $E \cap supp(q_i)$ by condition (6). It then easily follows that $\pi = \pi_E$ as desired. $\square$

Corollary 9.10 *If $\langle\langle U, N\rangle, \mathcal{C}(N)\rangle$ is a highly recursive extended nonmonotonic rule system such that $\mathcal{E}(\langle U, N\rangle) \neq \emptyset$, then there is an extension E of $\langle U, N\rangle$ and a set $B \in \Pi^1_1$ such that $E \leq_T B$.*

Proof: Kleene, see [Rog67], Theorem XLII(a), proved that for any nonempty Π^0_1 class $\mathcal{C}$, $\mathcal{C} \cap \{A : \exists_{B\in\Pi^1_1}(A \leq_T B)\} \neq \emptyset$. Our result then follows by letting $\mathcal{C} = [T^\star]$ for $T^\star$ corresponding to $\langle\langle U, N\rangle, \mathcal{C}(N)\rangle$ as constructed in the proof of Theorem 9.9. Thus there exists a path $\pi \in [T^\star]$ and a set $B \in \Pi^1_1$ such that $\pi \leq_T B$. But then $E_\pi \leq_T \pi \leq_T B$. $\square$

Corollary 9.11 *If $\langle\langle U, N\rangle, \mathcal{C}(N)\rangle$ is an recursive extended nonmonotonic rule system such that $\langle U, N\rangle$ has a unique extension E, then E is hyperarithmetic.*

Proof: Clearly, if a recursive tree T has a unique path, then this path is hyperarithmetic. Let $T^\star$ be the recursive tree constructed for $\langle\langle U, N\rangle, \mathcal{C}(N)\rangle$ in the proof of Theorem 9.9. Then $T^\star$ has exactly one infinite path (which is π_E, where E is the unique extension of $\langle U, N\rangle$. But $E \leq \pi_E$, and since π_E is hyperarithmetic, so is E, as the set of all hyperarithmetic sets is closed under Turing reducibility. $\square$

Our next result will show that Theorem 9.9 cannot be strengthened to ensure that for any recursive ENRS $\mathcal{S} = \langle U, N\rangle$ there exists a recursive tree T such that there is an effective, one-to-one degree-preserving correspondence between $\mathcal{E}(\mathcal{S})$ and $[T]$ (as is the case of Theorem 8.3 for nonmonotonic rule systems). We are indebted to the referee of the earlier version of this paper for suggesting the next theorem and for outlining an infinite injury priority argument for its proof. We shall not present the referee's proof but instead present a much simpler "wait-and-see" argument for the same result.

Theorem 9.12 *There exists a recursive extended nonmonotonic rule system $\mathcal{S} = \langle U, N\rangle$ such that there is no recursive tree $T \subseteq \omega^{<\omega}$ such that there is an effective one-to-one degree-preserving correspondence between $\mathcal{E}(\mathcal{S})$ and $[T]$.*

Proof: The universe of our extended nonmonotonic rule system will be the set of natural numbers $\omega = \{0, 1, 2, \ldots\}$. We will construct $\mathcal{S}$ so that $\mathcal{S}$ has no recursive extensions. This will ensure that there can be no effective 1:1 degree preserving correspondence between $\mathcal{E}(\mathcal{S})$ and $[T]$ if T has a recursive path.

It is well known that for every recursive tree T there is a primitive recursive tree T' such that T and T' have the same sets of infinite paths, i.e $[T] = [T']$. Thus it will be enough to show that there is no primitive recursive tree T' such that there is an effective 1:1 degree preserving correspondence between $\mathcal{E}(\mathcal{S})$ and $[T']$ for any primitive recursive tree T'.

Let $T_0, T_1, \ldots$ be such effective list of all primitive recursive trees contained in $\omega^{<\omega}$. Then, in order to ensure that there is no effective one-to-one degree-preserving correspondence between $\mathcal{E}(\mathcal{S})$ and $[T]$ for any recursive tree T, it is enough to meet the following set of requirements. (Here $<,>$ denotes some recursive bijection between $\omega \times \omega$ and ω.)

$R_{<e,n>}$

- **(i)** $[T_n]$ has a recursive element, or
- **(ii)** There is an extension E of $\mathcal{S}$ such that φ_e^E is not total, or
- **(iii)** There is an extension E of $\mathcal{S}$ such that $\varphi_e^E \notin [T_n]$.

Notice that since we construct $\mathcal{S}$ which has no recursive extensions, there cannot be a one-to-one, degree-preserving correspondence between $\mathcal{E}(\mathcal{S})$ and any $[T]$ satisfying (**i**).

In order to specify a recursive extended nonmonotonic rule system we must formally introduce codes for all the rules in N. We shall not construct the coding function explicitly. However, throughout the construction of $\mathcal{S}$, the rules of N will be constructed in such a way that it will be clear that each rule of N will have a recursive set as its set of constraints. It will then be obvious that an appropriate coding function can be constructed to show that $\mathcal{S}$ is a recursive extended nonmonotonic rule system.

The set of rules of N will be constructed in stages. We shall start by adding four classes of rules to N. The purpose of our first three classes of rules is to ensure that each extension E of $\mathcal{S}$ has exactly one even number in it. We shall ensure that no other rules of $\mathcal{S}$ has an even number either as a premise or as a conclusion. Thus remaining rules of $\mathcal{S}$ can have no effect on the membership of an even number in an extension. Let $Ev = \{0, 2, 4, \ldots\}$ denote the set of even numbers.

1. $\dfrac{: Ev}{2n}$ for all $n \geq 0$.

2. $\dfrac{: Ev \setminus \{2n\}}{2n}$ for all $n \geq 0$.

3. $\dfrac{2n, 2m :}{2p}$ for all m, n, p where $m \neq n$.

Let us consider the effect of these three collections of rules on an extension E of the $\mathcal{S}$. The rules in (1) imply that if there is no even number in E, then all even numbers are in $cl_E(\emptyset)$. Therefore every extension will have at least one even member. Next, the rules in (3) imply that if E contains at least two even numbers, then it contains all even numbers. However in that case the rules in (1) and (2) are blocked for E so they cannot be used to generate any even number into $cl_E(\emptyset)$. The rules in (3) cannot be used to generate elements in $cl_E(\emptyset)$ until at least two distinct even numbers have been derived in $cl_E(\emptyset)$. Since the rules in (1), (2), and (3) are the only rules which have even numbers as their conclusion, it would follow that there would be no even numbers in $cl_E(\emptyset)$ which contradicts the fact that E is an extension. Thus any extension must contain exactly one even number. Now if $2n$ is the only even number in E, then the rules of (2) show that $2n \in cl_E(\emptyset)$. Thus the rules in (1), (2) and (3) plus the fact that no other rules in N will have an even number as its conclusion imply that we will be able to decompose the set of extensions $\mathcal{E}(\mathcal{S})$ into pairwise disjoint sets $\mathcal{E}_0(\mathcal{S})$, $\mathcal{E}_2(\mathcal{S}), \ldots$ where $\mathcal{E}_{2n}(\mathcal{S}) = \{E \in \mathcal{E}(\mathcal{S}) : 2n \in E\}$.

Our idea is to use the set of extensions $\mathcal{E}_{2<e,n>}(\mathcal{S})$ to help us meet requirement $R_{<e,n>}$. For each $< e, n >$, we shall construct a set of rules $N_{2<e,n>}$ all of which have the set $Ev \setminus \{2 < e, n >\}$ contained in their constraint set. Note that if $k \neq < e, n >$, then such rules will all be blocked for any $E \in \mathcal{E}_{2k}(\mathcal{S})$ so that the set of rules $N_{2<e,n>}$ will have no effect on the possible extensions in $\mathcal{E}_{2k}(\mathcal{S})$. Thus the rules of the type (1), (2) and (3) plus the form of the rules in $N_{2<e,n>}$ allow us to break up the problem of meeting the requirements $R_{<e,n>}$ into a set of requirements which have no interaction with each other.

We shall construct the rules $N_{2<e,n>}$ in stages so that $\mathcal{E}_{2<e,n>}(\mathcal{S})$ consists of a single recursively enumerable, nonrecursive extension $E_{2<e,n>}$ such that either

$I_{2<e,n>}$: $\varphi_e^{E_{2<e,n>}}$ is not total, or

$II_{2<e,n>}$: $\varphi_e^{E_{2<e,n>}} \notin [T_n]$, or

$III_{2<e,n>}$: $[T_n]$ contains a recursive element.

Let *Odd* denote the set of odd numbers and partition *Odd* into three infinite recursive sets Q, S, and Z. Let A be some recursively enumerable nonrecursive, subset of ω and let f be a recursive injection whose range is A. Finally, let $\langle q_n : n \in \omega\rangle$, $\langle s_n : n \in \omega\rangle$ be increasing enumerations of Q and S. The set Z will be further partitioned below.

Our fourth class of rules will be partitioned into infinitely many classes $\{(4)_{2k}\}_{k\geq 0}$ where for all e and n, $(4)_{2<e,n>} \subseteq N_{2<e,n>}$.

$$(4)_{2<e,n>} \quad r_s^{2<e,n>} = \frac{: (Ev \setminus \{2<e,n>\}) \cup \{s_0\} \cup (S \setminus \{s_0, \ldots, s_{g(s)}\})}{q_{f(s)}}$$

for $s = 0, 1, \ldots$ where $g(s)$ is some recursive function which ensures that the codes $c(r_s^{2<e,n>})$ satisfy

$$c(r_0^{<e,n>}) < c(r_1^{<e,n>}) < c(r_2^{<e,n>}) < \ldots.$$

Note that since there are infinitely many rules of the form:

$$\frac{: (Ev \setminus \{2<e,n>\}) \cup \{s_0\} \cup (S \setminus \{s_0, \ldots, s_k\})}{q_{f(s)}}$$

we can easily construct a recursive function g with the required properties and hence ensure that that the set of rules in $(4)_{2<e,n>}$ will be a recursive set of rules.

The rules in the class $(4)_{2<e,n>}$ are designed to ensure that the extension $E_{2<e,n>}$ will be of the form

$$\{2<e,n>\} \cup \{q_i : i \in A\} \cup K$$

where K is some recursive subset of Z if we fail to satisfy either of the conditions $I_{2<e,n>}$ and $II_{2<e,n>}$. The exact role of this class of rules will become clear after we describe our attempt to satisfy conditions $I_{2<e,n>}$ and $II_{2<e,n>}$.

Our strategy for satisfying condition $I_{2<e,n>}$ or $II_{2<e,n>}$ is as follows. First we partition Z into infinitely many pairwise disjoint infinite recursive sets $Z_0, Z_1, \ldots$. For each i, let $Z_i = \{z_{0,i} : i < \omega\}$ be an increasing enumeration of Z_i. If we are successful in satisfying condition $I_{1<e,n>}$ or $II_{2<e,n>}$, then $E_{2<e,n>}$ will be of the form $\{2 < e, n >, s_0\} \cup P$ where P is some recursively enumerable nonrecursive subset of Z.

Given a finite sequence $\sigma = \langle \sigma_0, \sigma_1, \ldots, \sigma_p \rangle$ of zeros and ones, we shall regard σ as specifying a finite set $B_\sigma = \{i : i \leq p \wedge \sigma_i = 1\}$. We shall say that such σ is $2 < e, n >$-*compatible* if

$$\{2 < e, n >, s_0\} \subseteq B_\sigma \subseteq Z \cup \{2 < e, n >, s_0\}$$

Thus if σ is $2 < e, n >$-compatible, then $|\sigma| > \max(2 < e, n >, s_0)$. Let $\eta_0, \eta_1, \ldots$ be an effective list of all $2 < e, n >$-compatible sequences in which every $2 < e, n >$-compatible sequence occurs infinitely many times. Our idea is to process each finite sequence in this list in the stages of our construction of the set of rules $N_{2<e,n>}$.

Stage 0 of the construction of $N_{2<e,n>}$.
Suppose that the sequence η_0 is $\langle \sigma_0, \ldots, \sigma_p \rangle$. Let $\gamma_0^0, \gamma_1^0, \ldots$ be an effective enumeration of all $2 < e, n >$-compatible sequences of zeros and ones which extend η_0. Then consider the following sets of rules:

(A^0) $r_{i,0}^{2<e,n>} = \dfrac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_0 \setminus \{0, \ldots, p\})}{i}$

for each i such that $1 \leq i \leq p$ such that $\sigma_i = 1$.

(B^0) $b_0^{2<e,n>} = \dfrac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_0 \setminus \{0, \ldots, p\})}{s_0}$

(C^0) $a_{s,0}^{2<e,n>} = \dfrac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_0 \setminus (\{0, \ldots, p\} \cup \{z_{0,0}, \ldots z_{h_0(s),0}\}))}{z_{1+p+f(s),1}}$

for each $s \geq 0$. Here h_0 is a recursive function that ensures that

$$c(a_{0,0}^{2<e,n>}) < c(a_{1,0}^{2<e,n>}) < c(a_{2,0}^{2<e,n>}) < \ldots$$

so that the rules in (C^0) form a recursive set of rules.

Note that if $N_{2<e,n>}$ consisted only of the rules in $(4)_{2<e,n>}$ plus $(A^0), (B^0)$ and (C^0), then there would be a unique extension E in $\mathcal{E}_{2<e,n>}(\mathcal{S})$, namely,

$$E = \{2 < e, n >, s_0\} \cup B_{\eta_0} \cup \{z_{1+p+i,1} : i \in A\}.$$

That is, clearly, no element of $Z_0 \setminus B_{\eta_0}$ can be a member of an extension since there would be no rule in $N_{2<e,n>}$ whose conclusion is in $Z_0 \setminus B_{\eta_0}$. Thus the rules in $(A^0), (B^0)$, and (C^0) would force $B_{\eta_0} \cup \{z_{1+p+i,1} : i \in A\} \cup \{s_0\}$ in E. The presence of s_0 in E would ensure that none of the rules in $(4)_{2<e,n>}$ are E-applicable so that

$$E = \{2 < e, n >, s_0\} \cup B_{\eta_0} \cup \{z_{1+p+i,1} : i \in A\}$$

is the only possible extension. It is then straightforward to check that $E = cl_E(\emptyset)$ so that E is an extension in this case. Thus E is the unique extension in $\mathcal{E}_{2<e,n>}(\mathcal{S})$ in this case. Moreover, E is nonrecursive since A is a nonrecursive recursively enumerable set.

Our strategy is to attempt to add the rules in $(A^0), (B^0)$, and (C^0) to $N_{2<e,n>}$ in substages. At the substage 0 of stage 0, we add the rules in (A^0) and $((B^0)$ to $N_{2<e,n>}$. At substage $s > 0$ of stage 0, we add the rule $a^{2<e,n>}_{s-1,0}$ to $N_{2<e,n>}$ only if there is no $t \leq s$ such that if we compute $\varphi_{e,t}^{B_{\gamma_0^0}}(0), \ldots,$ $\varphi_{e,t}^{B_{\gamma_t^0}}(0)$ then for some $j \leq t$, $\varphi_{e,t}^{B_{\gamma_j^0}}(0)$ converges and $B_{\gamma_j^0} \subseteq \{2 < e, n >, s_0\} \cup B_{\eta_0} \cup \{z_{1+p+f(r),1} : r < t-1\}$. Here we say that $\varphi^C_{e,t}(x)$ converges if the e-th oracle machine with the input x and the oracle C gives an output in t or fewer steps. We also define the *use* of a convergent computation $\varphi^C_e(x)$, $use(C, e, x)$, to be the maximum of $\{0\} \cup \{h :\ h$ is a query to the oracle C in the computation of $\varphi^C_e(x)\}$. Clearly, if $\varphi^C_{e,s}(x)$ converges, then $use(C, e, x) \leq s$.

If we are successful in adding $a^{2<e,n>}_{s-1,0}$ to $N_{2<e,n>}$ at each substage $s > 0$ of stage 0, then we will have ensured that E as described above is the only extension of $\mathcal{E}_{2<e,n>}(\mathcal{S})$ and that $\varphi^E_e(0)$ is not defined. Thus we would satisfy the requirement $R_{<e,n>}$.

So suppose that s is the least substage of stage 0 at which we cannot add $a^{2<e,n>}_{s-1,0}$ to $N_{2<e,n>}$. Then we have two cases.

Case 1. There is some γ_j^0 where $j \leq s$ and

$$B_{\gamma_j^0} \subseteq \{2 < e, n >, s_0\} \cup B_{\eta_0} \cup \{z_{1+p+f(r),1} : r < s\}$$

where $\langle \varphi_{e,s}^{B_{\gamma_j^0}}(0) \rangle \notin T_n$. In this case we pick the least $k \in Z_0$ such that $k > \max(s, |\gamma_j^0| + 2)$ and the rule

$$\frac{: Ev \setminus \{2 < e, n >\}}{k}$$

has a code which is strictly larger than the code of any rule in (A^0), (B^0), or $\{a_{t,0}^{2<e,n>} : t < s\}$ and k is in the constraint set of all the rules in (A^0), (B^0), and $\{a_{t,0}^{2<e,n>} : t < s\}$. If k_0 is the least such k then we add

$$p_0 = \frac{: Ev \setminus \{2 < e, n >\}}{k_0}$$

to $N_{2<e,n>}$. Note that the rule p_0 ensures that $k_0 \in E$ for any extension E of $\mathcal{E}_{2<e,n>}(\mathcal{S})$ and has the effect of killing all the rules we have added to $N_{2<e,n>}$ up to this point at the stage 0 in the sense that they will not be E-applicable for any $E \in \mathcal{E}_{2<e,n>}(\mathcal{S})$. We then add the following set of rules to $N_{2<e,n>}$. Let $\gamma_j^0 = \langle \beta_0, \ldots, \beta_q \rangle$.

$(A^{0,1})$ $r_{i,0,1}^{2<e,n>} = \dfrac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_2 \setminus \{0, \ldots, q\})}{i}$
for each i such that $1 \leq i \leq q$ such that $\beta_i = 1$.

$(B^{0,1})$ $b_{0,1}^{2<e,n>} = \dfrac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_2 \setminus \{0, \ldots, q\}))}{s_0}$

$(C^{0,1})$ For each $r \geq 0$
$a_{r,0,1}^{2<e,n>} = \dfrac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_2 \setminus (\{0, \ldots, q\} \cup \{z_{0,q}, \ldots z_{h_1(r),2}\}))}{z_{k_0+f(r),3}}$
where h_1 is a recursive function which ensures that

$$c(a_{0,0,1}^{2<e,n>}) < c(a_{1,0,1}^{2<e,n>}) < c(a_{2,0,1}^{2<e,n>}) < \ldots$$

so that the rules in $(C^{0,1})$ form a recursive set.

In this case we can argue almost exactly as we did above that there is a unique extension in $\mathcal{E}_{2<e,n>}(\mathcal{S})$, namely

$$E_{2<e,n>} = \{2 < e, n >, s_0, k_0\} \cup \{z_{k_0+i,3} : i \in A\} \cup B_{\gamma_j^0}.$$

Note that since $i \leq z_{i,3}$ and hence $k_0 \leq z_{k_0+i,3}$ for all i,

$$E_{2<e,n>} \cap \{0, \ldots, k_0 - 1\} = B_{\gamma_j^0} \cap \{0, \ldots, k_0 - 1\}.$$

Since $use(B_{\gamma_j^0}, e, 0) < k_0$, it follows that $\varphi_e^{B_{\gamma_j^0}}(0) = \varphi_e^{E_{2<e,n>}}(0)$. Thus we have ensured that since $\langle \varphi_e^{B_{\gamma_j^0}}(0) \rangle \notin T_n$, either $\varphi_e^{E_{2<e,n>}}$ is not total or $\varphi_e^{E_{2<e,n>}} \notin [T_n]$.

In the first two situations, i.e. where we are able to add $a_{s,0}^{2<e,n>}$ to $N_{2<e,n>}$ for each $s \geq 0$ or where we fail to do this and end up in case 1, we do not add any more rules to $N_{2<e,n>}$, and we do not go to the next stage. In both these situations $\mathcal{E}_{2<e,n>}(\mathcal{S})$ will consist of a single, recursively enumerable nonrecursive extension and one of the conditions $I_{2<e,n>}$ or $II_{2<e,n>}$ will hold.

Case 2. We assume that the assumptions of Case 1 do not hold.

In this case, there is a γ_j^0 extending η_0 such that $\varphi_e^{B_{\gamma_j^0}}(0)$ is defined and $\langle \varphi_e^{B_{\gamma_j^0}}(0) \rangle \in T_n$. Then we pick the least $k \in Z_0$ such that $k > \max(s, |\gamma_j^0| + 2)$ and the rule

$$\frac{: Ev \setminus \{2 < e, n >\}}{k}$$

has a code which is strictly larger than the code of any rule in (A^0), (B^0), or $\{a_{t,0}^{2<e,n>} : t < s\}$ and k belongs to the constraint set of all the rules in (A^0), (B^0), and $\{a_{t,0}^{2<e,n>} : t < s\}$. If k_0 is the least such k, then we add the rule

$$p_0 = \frac{: Ev \setminus \{2 < e, n >\}}{k_0}$$

to $N_{2<e,n>}$ and go onto the Stage 1. As before the effect of adding of the rule p_0 to $N_{2<e,n>}$ is that it ensures that k_0 belongs to every extension of $\mathcal{E}_{2<e,n>}(\mathcal{S})$ and hence none of the rules added to $N_{2<e,n>}$ up to this point in stage 0 are E-applicable for any $E \in \mathcal{E}_{2<e,n>}(\mathcal{S})$. Thus in Case 2, we essentially negate the effect of all rules added to $N_{2<e,n>}$ at stage 0 except for p_0 for any extension of $\mathcal{E}_{2<e,n>}(\mathcal{S})$. The rule p_0 forces k_0 to be in all extensions of $\mathcal{E}_{2<e,n>}(\mathcal{S})$. We also know that there is at least one $2 < e, n >$-compatible extension γ of η_0 such that $\varphi_e^{B_\gamma}(0)$ is defined, $\langle \varphi_e^{B_\gamma}(0) \rangle \in T_n$, and $use(B_\gamma, e, 0) < k_0$. Thus if δ_0 is a sequence of length $k_0 + 1$ such that $B_{\delta_0} = \{k_0\} \cup B_\gamma$, then $\varphi_e^{B_{\delta_0}}(0)$ is defined, $\langle \varphi_e^{B_{\delta_0}}(0) \rangle \in T_n$, δ_0 is $2 < e, n >$-compatible sequence extending η_0, and $use(B_{\delta_0}, e, 0) < |\delta_0|$ so that $\varphi_e^{B_{\delta_0}}(0) = \varphi_e^{B_\eta}(0)$ for all η extending δ_0.

The structure of stage s of our construction is essentially the same as stage 0. That is, if we get to stage s, we assume that we defined integers $k_0 < \ldots < k_{s-1}$ where for each $i < s$, $k_i \in Z_i$, and we have added rules

$$p_i = \frac{: Ev \setminus \{2 < e, n >\}}{k_i}$$

which have the effect of ensuring that any other rules added to $N_{2<e,n>}$ at stages $0, \ldots, s-1$ are not applicable for any $E \in \mathcal{E}_{2<e,n>}(\mathcal{S})$. Since we added only finitely many rules to $N_{2<e,n>}$ at stages $0, \ldots, s-1$, we can compute the maximum of the codes of all the rules added at the stages $0, \ldots, s-1$. We denote this number by m_s.

Stage $s > 0$ of the construction of $N_{2<e,n>}$.

Suppose $\eta_s = \langle \sigma_0, \ldots, \sigma_p \rangle$.

If there is a k_i with $i < s$ such that $k_i + 1 \leq |\eta_s|$ and $k_i \notin B_{\eta_s}$, then let k_s be the least $k \in Z_s$ such that $k > k_{s-1}$ and the code of the rule

$$p_k = \frac{: Ev \setminus \{2 < e, n >\}}{k}$$

is strictly bigger than the code of any rule added to $N_{2<e,n>}$ at stages $0, \ldots, s-1$. In this case we add p_{k_s} to $N_{2<e,n>}$ and go directly to the stage $s+1$. Otherwise, proceed as follows:

Let $\gamma_0^s, \gamma_1^s, \ldots$ will be an effective list of all $2 < e, n >$-compatible sequences γ such that the length of γ is greater or equal than $k_{s-1} + 1$, γ extends η_s, and $\{k_0, \ldots, k_{s-1}\} \subseteq B_\gamma$.

Consider the following sets of rules:

(A^s) $r_{i,s}^{2<e,n>} = \dfrac{:(Ev \setminus \{2<e,n>\}) \cup (Z_0 \setminus (\{0,\ldots,p\} \cup \{z_{0,s},\ldots,z_{u_i,s}\}))}{i}$
for each i such that $1 \leq i \leq p$ such that $\sigma_i = 1$. Here $u_0,\ldots,u_p$ are chosen so that $m_s < c(r_{i,s}^{2<e,n>})$.

(B^s) $b_s^{2<e,n>} = \dfrac{:(Ev \setminus \{2<e,n>\}) \cup (Z_s \setminus (\{0,\ldots,p\} \cup \{z_{0,s},\ldots,z_{v,s}\}))}{s_0}$
where v is selected so that $c(b_s^{2<e,n>})$ is larger than the code of any rule in A^s. Note A^s is not empty since s_0 and $2<e,n>$ are B_{η_s}

(C^s) $a_{t,s}^{2<e,n>} = \dfrac{:(Ev \setminus \{2<e,n>\}) \cup (Z_s \setminus (\{0,\ldots,p\} \cup \{z_{0,s},\ldots z_{h_{2s}(t),s}\}))}{z_{1+k_{s-1}+f(t),s+1}}$
for each $t \geq 0$. Here h_{2s} is a recursive function that ensures that

$$c(b_s^{2<e,n>}) < c(a_{0,s}^{2<e,n>}) < c(a_{1,s}^{2<e,n>}) < c(a_{2,s}^{2<e,n>}) < \ldots$$

so that the rules in (C^s) form a recursive set of rules.

We can then argue exactly like in stage 0 that if $N_{2<e,n>}$ consisted of the rules in $(4)_{2<e,n>}$ plus the rules added at the stages $0,\ldots,s-1$ plus the rules in (A^s), (B^s), and (C^s), then $\mathcal{E}_{2<e,n>}(\mathcal{S})$ consists of a unique extension E where

$$E = \{2<e,n>, s_0, k_0, \ldots, k_{s-1}\} \cup B_{\eta_0} \cup \{z_{1+k_{s-1}+i,s+1} : i \in A\}$$

Once again we attempt to add the rules in (A^s), (B^s), and (C^s) in substages. At substage 0 of stage s, we add the rules (A^s) and (B^s) to $N_{2<e,n>}$. At substage $t>0$ of stage s, we add $a_{t-1,s}^{2<e,n>}$ only if there is no $j \leq t$ such that if we compute $\varphi_{e,t}^{B_{\gamma_j^s}}(0),\ldots,\varphi_{e,t}^{B_{\gamma_j^s}}(s)$, then $\varphi_{e,t}^{B_{\gamma_j^s}}(i)$ converges for $i=0,\ldots,s$,

$$B_{\gamma_j^s} \subseteq \{2<e,n>, s_0, k_0, \ldots k_{s-1}\} \cup B_{\eta_s} \cup \{z_{f(q),s+1} : q < r-1\},$$

and $k_l \in B_{\gamma_j^s}$ for $l < s$.

If we are successful in adding $a_{t-1,s}^{2<e,n>}$ to $N_{2<e,n>}$ at each substage t of stage s, then we will have ensured that E as described above is the only extension of $\mathcal{E}_{2<e,n>}(\mathcal{S})$. Moreover, we will have ensured that it is not the case that $\varphi_e^E(i)$ is defined for all $i \leq s$. Thus we would automatically satisfy requirement $R_{<e,n>}$.

If we are not successful in adding in adding $a_{r,s}^{2<e,n>}$ to $N_{2<e,n>}$ for all $r > 0$, let t be the least substage for which we cannot add $a_{t-1,s}^{2<e,n>}$ to $N_{2<e,n>}$. As before we have two cases.

Case 1: There is a sequence γ_j^s which extends η_s such that $j \leq t$, $\varphi_{e,t}^{B_{\gamma_j^s}}(0), \ldots, \varphi_{e,t}^{B_{\gamma_j^s}}(s)$ are all defined, $\langle \varphi_{e,t}^{B_{\gamma_j^s}}(0), \ldots, \varphi_{e,t}^{B_{\gamma_j^s}}(s) \rangle \notin T_n$,

$$B_{\gamma_j^s} \subseteq \{2 < e, n >, k_0, \ldots, k_{s-1}\} \cup B_{\eta_s} \cup \{z_{f(r),s} : r < t-1\},$$

and $k_i \in B_{\gamma_j^s}$ if $i < s$. In this case we pick the least $k \in Z_s$ such that the rule

$$\frac{: Ev \setminus \{2 < e, n >\}}{k}$$

has a code which is strictly larger than the code of any rule added to $N_{2<e,n>}$ at stages $0, \ldots, s$ up to this point, $k > \max(t, k_{s-1}, |\gamma_j^s| + 2)$, and k is in the constraint set of all rules in (A^s), (B^s), and $\{a_{r,s}^{2<e,n>} : r < t\}$. If k_s is the least such k, then we add

$$p_s = \frac{: Ev \setminus \{2 < e, n >\}}{k_s}$$

to $N_{2<e,n>}$. Note that the rule p_s ensures that $k_s \in E$ for any extension in $\mathcal{E}_{2<e,n>}(\mathcal{S})$ and hence none of the rules we have added to $N_{2<e,n>}$ at the stage s other than p_s will be applicable for any $E \in \mathcal{E}_{2<e,n>}(\mathcal{S})$. We then add the following set of rules to $N_{2<e,n>}$. Let $\gamma_j^s = \langle \beta_0, \ldots \beta_q \rangle$.

$(A^{s,1})$ $$r_{i,s,1}^{2<e,n>} = \frac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_{s+2} \setminus (\{0, \ldots, q\} \cup \{z_{0,s+2}, \ldots, z_{u_i,s+2}\}))}{i}$$
for each i such that $1 \leq i \leq q$ such that $\beta_i = 1$ where u_i is chosen so that $c(r_{i,s,i}^{2,<e,n>})$ is larger than the code of any rule added to $N_{2<e,n>}$ at stages $0, \ldots, s$ up to this point.

$(B^{s,1})$ $$b_{s,1}^{2<e,n>} = \frac{: (Ev \setminus \{2 < e, n >\}) \cup (Z_{s+2} \setminus (\{0, \ldots, q\} \cup \{z_{0,s+2}, \ldots, z_{v,s+2}\}))}{s_0}$$
where v is chosen so that $c(b_{s,1}^{2<e,n>})$ is larger than the code of any rule in $(A^{s,1})$. Note $A^{s,1}$ is not empty since s_0 and $2 < e, n >$ are in $B_{\gamma_j^s}$.

$(C^{s,1})$ For each $r \geq 0$,

$$a_{r,s,1}^{2<e,n>} = \frac{:(Ev \setminus \{2<e,n>\}) \cup (Z_{s+2} \setminus (\{0,\ldots,q\} \cup \{z_{0,s+2},\ldots z_{h_{2s+1}(r),s+2}\}))}{z_{1+k_s+f(r),s+3}}$$

where h_{2s+1} is a recursive function which ensures that

$$c(b_{s,1}^{2<e,n>} < c(a_{0,s,1}^{2<e,n>}) < c(a_{1,s,1}^{2<e,n>}) < c(a_{2,s,1}^{2<e,n>}) < \ldots$$

so that the rules in $(C^{s,1})$ form a recursive set.

In this case we can argue as we did in stage 0 that there is a unique extension in $\mathcal{E}_{2<e,n>}(\mathcal{S})$, namely

$$E_{2<e,n>} = \{2<e,n>, s_0, k_0, \ldots, k_s\} \cup \{z_{1+k_s+i,s+3} : i \in A\} \cup B_{\gamma_j^s}.$$

Moreover since $\langle \varphi_e^{B\gamma_j^s}(0), \ldots, \varphi_e^{B\gamma_j^s}(s)\rangle \notin T_n$, then it will be the case that either $\varphi_e^{E_{2<e,n>}}$ is not total or $\varphi_e^{E_{2<e,n>}} \notin [T_n]$.

Once again, if we are in the first two situations at stage s, i.e. where we are able to add $a_{t,s}^{2<e,n>}$ to $N_{2<e,n>}$ for each $t \geq 0$ or where we fail to do this and end up in case 1, we do not add any more rules to $N_{2<e,n>}$, and we do not go to the next stage. In both these situations, $\mathcal{E}_{2<e,n>}(\mathcal{S})$ will consist of a single recursively enumerable nonrecursive extension $E_{2<e,n>}$ and either $\varphi_e^{E_{2<e,n>}}$ is not total or $\varphi_e^{E_{2<e,n>}} \notin [T_n]$.

Case 2: The assumptions of Case 1 do not hold.
In this case there must be a sequence γ_j^s extending η_s such that

(i) $|\gamma_j^s| \geq k_{s-1}+1$,

(ii) $\{2<e,n>, s_0, k_0, \ldots, k_{s-1}\} \subseteq B_{\gamma_j^s}$,

(iii) $\varphi_{e,t}^{B_{\gamma_j^s}}(i)$ is defined for $i = 0, \ldots, s$, and

(iv) $\langle \varphi_{e,t}^{B_{\gamma_j^s}}(0), \ldots, \varphi_{e,t}^{B_{\gamma_j^s}}(s)\rangle \in T_n$.

Then we pick the least $k \in Z_s$ such that $k > max(t, k_{s-1}, |\gamma_j^s|+2)$, the rule

$$\frac{:Ev \setminus \{2<e,n>\}}{k}$$

has a code which is strictly larger than the code of any rule that we have added to $N_{2<e,n>}$ at stages $0, \ldots, s$ up to this point. If k_s is the least such k, then we add

$$p_s = \frac{: Ev \setminus \{2 < e, n >\}}{k_s}$$

to the set $N_{2<e,n>}$ and go to stage $s+1$. Note that in this case, if we let δ_s be the string of length k_s such that $B_{\delta_s} = \{k_s\} \cup B_{\gamma_j^s}$, then $\{2 < e, n >, s_0, k_0, \ldots, k_s\} \subseteq B_{\delta_s}$, δ_s extends η_s, and $\varphi_e^{B_{\delta_s}}(i) = \varphi_e^{B_{\gamma_j^s}}(i)$ for $i \leq s$ since for each such i, B_{δ_s} and $B_{\gamma_j^s}$ agree up to $use(B_{\gamma_j^s}, e, i)$. Thus, since $|B_{\delta_s}| - 1 \geq use(B_{\gamma_j^s}, e, i)$ for all $i \leq s$, any sequence η which extends δ_s will have $\varphi_e^{B_\delta}(i) = \varphi_e^{B_{\gamma_j^s}}(i)$ for all $i \leq s$.

Finally we consider the situation where we complete all the stages $s \geq 0$. In this situation, it is easy to see that we add a *finite* number of rules to $N_{2<e,n>}$ at each stage s and our construction ensures that the codes of all the rules added to $N_{2<e,n>}$ at stage $s+1$ are larger than the codes of rules added to $N_{<e,n>}$ at stage s. This fact ensures that $N_{2<e,n>}$ is a recursive set of rules. The rules $p_0, p_1, \ldots$ force that $K = \{k_0, k_1 \ldots\}$ is a subset of any extension $E \in \mathcal{E}_{2<e,n>}(\mathcal{S})$ and no other rules added to $N_{2<e,n>}$ during stages $0, 1, \ldots$ are E-applicable. The rules in $(4)_{2<e,n>}$, then show that there is a unique extension $E_{2<e,n>}$ of $\mathcal{E}_{2<e,n>}(\mathcal{S})$ where

$$E_{2<e,n>} = \{2 < e, n >\} \cup K \cup \{q_i : i \in A\}.$$

It is easy to see that our construction ensures that $k_0 < k_1 < \ldots$ is an recursively enumerable increasing sequence so that K is a recursive set and therefore $E_{2<e,n>}$ is a recursively enumerable set nonrecursive set.

Moreover, in this case, we know that for any s and any $2 < e, n >$-compatible sequence $\langle \eta_j \rangle_{j \in \omega}$ such that $k_j \in B_{\eta_j}$ whenever $k_i + 1 < |\eta_j|$, there is a $2 < e, n >$-compatible sequence γ extending η_j such that

(a) $\varphi_e^{B_\gamma}(i)$ is defined for $i = 0, \ldots, s$,

(b) $\langle \varphi_e^{B_\gamma}(0), \ldots, \varphi_e^{B_\gamma}(s) \rangle \in T_n$,

(c) $k_i \in B_\gamma$ whenever $k_i + 1 \leq |\gamma|$, and

(d) $use(B_\gamma, e, i) \leq |\gamma| - 1$ for $i = 1, \ldots, s$.

This fact allows us to define an effective infinite sequence of strings of 0 and 1's, $\gamma_0 \sqsubseteq \gamma_1 \sqsubseteq \ldots$ as follows. Let $\gamma_0 = \langle \beta_0, \ldots, \beta_{\max(s_0, 2<e,n>)} \rangle$ be a string of 0's and 1's such that $\beta_i = 1$ iff $i \in \{s_0, 2 < e, n >\}$. By our construction, we can find a $2 < e, n >$-compatible string γ_1 extending γ_0 such that $\varphi_e^{B_{\gamma_1}}(0)$ is defined, $\langle \varphi_e^{B_{\gamma_1}}(0) \rangle \in T_n$, $use(B_{\gamma_{1,e,0}}) \leq |\gamma_1| - 1$, and $k_i \in B_{\gamma_1}$ whenever $k_i + 1 \leq |\gamma_1|$. Now assume that we have defined a sequence of $2 < e, n >$-compatible strings

$$\gamma_0 \sqsubseteq \gamma_1 \sqsubseteq \ldots \sqsubseteq \gamma_s$$

such that for all $j \leq s$,

(i) γ_j is $2 < e, n >$-compatible,

(ii) $k_i \in B_{\gamma_j}$ whenever $k_i + 1 \leq |\gamma_j|$,

(iii) $\varphi_e^{B_{\gamma_j}}(i)$ is defined for $i = 0, \ldots, j - 1$,

(iv) $\langle \varphi_e^{B_{\gamma_j}}(0), \ldots, \varphi_e^{B_{\gamma_j}}(j-1) \rangle \in T_n$, and

(v) $use(B_{\gamma_j}, e, i) \leq |\gamma_j| - 1$ for all $i \leq j - 1$.

Then by our construction we can find a $2 < e, n >$-compatible string γ_{s+1} extending γ_s such that

(a) $k_i \in B_{\gamma_{s+1}}$ whenever $k_i + 1 \leq |\gamma_{s+1}|$,

(b) $\varphi_e^{B_{\gamma_{s+1}}}(i)$ is defined for $i = 0, \ldots, s$,

(c) $\langle \varphi_e^{B_{\gamma_{s+1}}}(0), \ldots, \varphi_e^{B_{\gamma_{s+1}}}(s) \rangle \in T_n$, and

(d) $use(B_{\gamma_{s+1}}, e, i) \leq |\gamma_{s+1}| - 1$ for all $i \leq s$.

It then follows that if we define $\pi_s = \varphi_e^{B_{\gamma_{s+1}}}(s)$, then $\langle \pi_0, \pi_1, \ldots \rangle$ is an infinite recursive path through T_n and hence $[T_n]$ has a recursive element.

It is easy to see that $\{N_{2<e,n>}\}_{e,n\in\omega}$ is an effective sequence of recursive sets and hence with an appropriate coding function c one can easily show that $N = \bigcup_{e,n\in\omega} N_{2<e,n>}$ is also recursive. Thus $\mathcal{S} = \langle \omega, N\rangle$ is a recursive ENRS for which there is no recursive tree $T \subseteq \omega^{<\omega}$ such that there is an effective 1:1 correspondence between $\mathcal{E}(\mathcal{S})$ and $[T]$. $\Box$

We note that the infinite injury priority argument suggested by a referee constructs a recursive ENRS $\mathcal{S}$ satisfying Theorem 9.12 which has a unique extension. However we shall not give that construction here. We do note that the system $\mathcal{S}$ we constructed in Theorem 9.12 has the property that all the extensions of $\mathcal{S}$ have the same Turing degree, namely the degree of A.

10 $L_{\omega_1,\omega}$ characterization of extensions of extended nonmonotonic rule systems

In this section we will give a logical characterization of extensions for certain extended nonmonotonic rule systems. This characterization uses the infinitary logic $L_{\omega_1,\omega}$. The results we obtain here are similar to ones found in [MNR92a].

We will be interested in characterization of extensions of extended nonmonotonic rule systems $\langle U, N\rangle$ where both U and N are denumerable. Notice that for (nonextended) nonmonotonic rule systems, N is denumerable if U is denumerable. But now, for extended nonmonotonic rule systems, it is no longer the case that U being denumerable forces N to be denumerable. Indeed in such cases we can only conclude that $|N| \leq 2^{|U|}$. We will consider only the case when $|N| \leq \omega$.

Our point of departure is the observation that although the restraints of the rules are now possibly infinite, the S-proofs and proof schemes are still finite. Moreover, the relationship $\prec$ between proof schemes is still well-founded. Thus there is a minimal proof scheme of $x \in U$ below each proof scheme for x. However there is a subtle difference between the present situation, extended nonmonotonic rule systems, and the previous one for nonmonotonic rule systems with rules admitting only finite restraints. In this latter case, we not only could find a minimal proof scheme for each $a \in U$ which is the conclusion of some minimal proof scheme (as is also the case in the present situation) but in addition, given a support Z of a proof scheme for a, we could find a minimal proof scheme p' for a such that $supp(p') = Z'$ where $Z' \subseteq Z$ and Z' is a minimal set with this property. This is no longer the case for extended nonmonotonic rule systems.

Example 10.1 *Let* $U = \{b_0, b_1, \ldots, \}$ *and define* $r_i = \frac{:Z_i}{b_0}$ *where* $Z_i = \{b_{i+1}, b_{i+2}, \ldots\}$. *Then each* $\langle\langle b_0, r_i, Z_i\rangle\rangle$ *is proof scheme for* b_0. *Each* Z_i *is a support for one such scheme, but there is no proof scheme with empty support for* b_0.

However, the fundamental result for proof schemes (Proposition 6.2) is true, and we state a version of it suitable for our considerations.

Proposition 10.1 *Let* $\langle U, N\rangle$ *be an extended nonmonotonic rule system, and let* $a \in U$. *Let* M *be an extension of* $\langle U, N\rangle$.

1. *Let* s *be a proof scheme for* a *and suppose* $supp(p) = S$. *Then if* $S \cap M = \emptyset$, $a \in M$.
2. *Conversely, if* $a \in M$, *then there exists a proof scheme* s *such that* $supp(s) \cap M = \emptyset$.

Proof: Part 1 is proved by induction on the length of scheme s. Part 2 is proved by induction on the length of M-proofs of elements of M. □

In the case we are considering in this section where we assume that both U and N are denumerable, the set of all proof schemes for $\langle U, N\rangle$ is denumerable. Thus, for every $a \in U$ the set S_a of all proof schemes with conclusion a is denumerable. Consequently, the set $\mathcal{Z}_a$ of supports of all proof schemes in S_a is denumerable.

Now for every $u \in U$, let c_u be a new constant. Let $\mathcal{L}_\infty$ be the propositional language of $\mathcal{L}\omega_1, \omega$ generated by all constants c_u. Thus $\mathcal{L}_\infty$ is the least language containing all constants c_u and closed under negation and denumerable disjunctions and denumerable conjunctions.

We define the satisfaction relation $\models$ between subsets of U and formulas of $\mathcal{L}_\infty$ as follows:

Definition 10.2 1. $M \models c_u$ if $u \in M$

2. $M \models \neg\psi$ if $M \not\models \psi$

3. $M \models \bigwedge_{i\in I} \psi_i$ if for all $i \in I$ $M \models \psi_i$

4. $M \models \bigvee_{i\in I} \psi_i$ if for some $i \in I$ $M \models \psi_i$

Given $Z \subset U$, define a formula out_Z as

$$\bigwedge_{u\in Z} \neg c_u.$$

Given $a \in U$, let ϑ_a denote the formula

$$\bigvee_{Z\in\mathcal{Z}_a} out_Z.$$

Since $\mathcal{Z}_a$ is denumerable for every $a \in U$, ϑ_a is in $\mathcal{L}_\infty$ for each $a \in U$. Next we let $char_a$ denote the formula

$$c_a \Leftrightarrow \vartheta_a.$$

Finally, we let $char_{U,N}$ denote the formula

$$\bigwedge_{a\in U} char_a.$$

We then have the following logical characterization of extensions of $\langle U, N\rangle$.

Theorem 10.3 *Let $\langle U, N\rangle$ be an extended nonmonotonic rule system such that both U and N are denumerable and let $char_{U,N}$ be the sentence of $\mathcal{L}_\infty$ defined above. Then $M \subseteq U$ is an extension of $\langle U, N\rangle$ if and only if $M \models char_{U,N}$.*

Proof: We use Propositions 6.2 and 10.1. First, suppose that M is an extension of $\langle U, N\rangle$. Then if $a \in M$, there exists some proof scheme p with conclusion a which is admitted by M. Thus if $Z = supp(p)$, then $M \cap Z = \emptyset$ and hence $M \models out_Z$. Therefore $M \models \bigvee_{Z \in \mathcal{Z}_a} out_Z$. Moreover $M \models c_a$ (since $a \in M$). Therefore $M \models char_a$. Now if $a \notin M$, then since M is an extension of $\langle U, N\rangle$, $M \cap supp(s) \neq \emptyset$ for every proof scheme s for a. Thus if $Z = supp(s)$ where s is a proof scheme for a, then $M \models \neg out_Z$. It then follows that the formula $\bigvee_{Z \in \mathcal{Z}_a} out_Z$ is false in M. But c_a is also false in M and hence $M \models char_a$. Thus $M \models char_a$ for all $a \in U$ and hence $M \models char_{U,N}$.

Conversely, if $M \models char_{U,N}$, then for all $a \in U$, $M \models char_a$. By analyzing the form of $char_a$, it is easy to see that whenever $a \in M$, there is some proof scheme for a which is admitted by M, and when $a \notin M$, then there is no proof scheme for a which is admitted by M. Therefore M is an extension of $\langle U, N\rangle$. $\square$

References

[Apt90] K. Apt. Logic programming, In: J. van Leeuven, ed, Handbook of Theoretical Computer Science, pages 493–574, MIT Press, Cambridge, MA", 1990.

[AvE82] K.R. Apt and M.H. van Emden. Contributions too the theory of logic programming. *Journal of the ACM.*, 29:841–862, 1982.

[Bat89] A. Batarekh. Topological Aspects of Logic Programming. *Ph. D. dissertation*, University of Syracuse, 1989.

[BF91] N. Bidoit and Ch. Froidevaux. General logical databases and programs: Default logic semantics and stratification. *Information and Computation*, 91:15–54, 1991.

[Bir35] G. Birkhoff. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society*, 31:433–454, 1935.

[BBS89] H.A. Blair, A.L. Brown, V.S. Subrahmanian. Monotone Logic Programming. In: *Intentional Logics for Programming*, L. Farinas del Cerro and M. Penttonen, eds, pages 1–22, Oxford Univ. Press, 1992.

[Che80] B.F. Chellas. *Modal logic, an introduction*. Cambridge University Press, 1980.

[Cla78] K.L. Clark. Negation as failure. In H. Gallaire and J. Minker, editors, *Logic and data bases*, pages 293–322. Plenum Press, 1978.

[Clo85] P. Clote. On recursive trees with a unique infinite branch. *Proceedings of the American Mathematical Society*, 93:335–342, 1985.

[Doy79] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12:231–272, 1979.

[Ell74] E. Ellentuck A new proof that analytic sets are Ramsey. *Journal of Symbolic Logic* 39:163-165, 1974.

[Fer91] A. Ferry. Enriched Nonmonotonic Rule Systems *M.Sc. Thesis* University of Kentucky, 1991.

[Fer92] A. Ferry. A topological characterization of stable and minimal model classes of propositional logic programs. *Annals of Mathematics and Artificial Intelligence*, To appear.

[Fer93] A. Ferry. Characterizations and properties of classes associated with logic programming semantics. To appear.

[GL88] M. Gelfond and V. Lifschitz. The stable semantics for logic programs. In R. Kowalski and K. Bowen, editors, *Proceedings of the 5th international symposium on logic programming*, pages 1070–1080, Cambridge, MA., 1988. MIT Press.

[GL90] M. Gelfond and V. Lifschitz. Logic programs with classical negation. In D. Warren and P. Szeredi, editors, *Proceedings of the 7th international conference on logic programming*, pages 579–597, Cambridge, MA., 1990. MIT Press.

[JS72a] C.G. Jockusch and R.I. Soare. Degrees of members of π_1^0 classes. *Pacific Journal of Mathematics*, 40:605–616, 1972.

[JS72b] C.G. Jockusch and R.I. Soare. π_1^0 classes and degrees of theories. *Transactions of American Mathematical Society*, 173:33–56, 1972.

[McD82] D. McDermott. Nonmonotonic logic II: nonmonotonic modal theories. *Journal of the ACM*, 29:33–57, 1982.

[MD80] D. McDermott and J. Doyle. Nonmonotonic logic I. *Artificial Intelligence*, 13:41–72, 1980.

[MNR90] W. Marek, A. Nerode, and J.B. Remmel. Nonmonotonic rule systems I. *Annals of Mathematics and Artificial Intelligence*, 1:241–273, 1990.

[MNR92a] W. Marek, A. Nerode, and J.B. Remmel. Nonmonotonic rule systems II. *Annals of Mathematics and Artificial Intelligence*, 5:229–263, 1992.

[MNR92b] W. Marek, A. Nerode, and J.B. Remmel. How complicated is the set of stable models of a recursive logic program? *Annals of Pure and Applied Logic*, 33:229–263, 1992.

[MNR94a] W. Marek, A. Nerode, and J. B. Remmel. A Context for Belief Revision: Forward-Chaining Normal Nonmonotonic Rule Systems. Annals of Pure and Applied Logic 67(1994) pp. 269-324.

[MNR94b] W. Marek, A. Nerode, and J. B. Remmel. The stable models of a predicate logic program. *Journal of Logic Programming* 21(1994) pp. 129-154.

[MNR94c] W. Marek, A. Nerode, and J. B. Remmel. Rule systems, well-orderings and forward chaining. *In preparation.*, 1993.

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

[MT89a] W. Marek and M. Truszczyński. Relating autoepistemic and default logics. *Proceedings of the 1st international conference on principles of knowledge representation and reasoning*, KR '89", pages 276–288, 1989.

[MT91] W. Marek and M. Truszczyński. Autoepistemic logic. *Journal of the ACM*, 38:588 – 619, 1991.

[MT89] W. Marek and M. Truszczyński. On reasonings by a fully introspective, intelligent, agent. In *Proceedings of the international conference on computing and information*, pages 417–419, Amsterdam, 1989. North-Holland.

[MT93] W. Marek and M. Truszczyński. *Nonmonotonic logics; context-dependent reasoning.* Berlin: Springer-Verlag, 1993.

[RDB89] M. Reinfrank, O. Dressler, and G. Brewka. On the relation between truth maintenance and non-monotonic logics. In *Proceedings of IJCAI-89*, pages 1206–1212, San Mateo, CA., 1989. Morgan Kaufmann.

[Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[Rog67] H. Rogers. *Theory of recursive functions and effective computability.* McGraw-Hill, 1967.

[Sch52] J. Schmidt. Uber die Rolle der transfinites Schlussweisen in einer algemeiner Algebra. *Mathematische Nachrichten* 7:165-182, 1952.

[Tar56] A. Tarski. *Logic, semantics, metamathematics.* Oxford University Press, Oxford, 1956.

[VGRS91] A. Van Gelder, K.A. Ross and J.S. Schlipf. Unfounded sets and well-founded semantics for general logic programs. Journal of the ACM 38(1991).

[YBB92] F. Yang, H. Blair, and A. Brown. Programming in default logic. University of Syracuse, 1992.