Some effectively infinite classes of enumerations

Sergey Goncharov
Institute of Mathematics, Novosibirsk, Russian Federation

Alexander Yakhnis
Mathematical Sciences Institute, Cornell University, Ithaca, NY 14853, USA

Vladimir Yakhnis
IBM Corporation, Endicott, NY 13760, USA

Communicated by A. Nerode
Received 1 December 1990
Revised 18 November 1992

Abstract


This research partially answers the question raised by Goncharov about the size of the class of positive elements of a Roger's semilattice. We introduce a notion of effective infinity of classes of computable enumerations. Then, using finite injury priority method, we prove five theorems which give sufficient conditions to be effectively infinite for classes of all enumerations without repetitions, positive undecidable enumerations, negative undecidable enumerations and all computable enumerations of a family of r.e. sets. These theorems permit to strengthen the results of Pour-El, Pour-El and Howard, Ershov and Khutoretskii about existence of enumerations without repetitions and positive undecidable enumerations.

1. Introduction

Our purpose is to study the classical families of all r.e. sets and all p.r. functions with respect to effective infinity of various classes of computable enumerations. Countable classes of objects may have different degrees of infinity. We may first show that a class is not finite. This is the classical version of infinity. Another possibility is to construct an infinite sequence without repetitions consisting of objects from the class. This yields more information than the...
classical version and it is more constructive. A still stronger way to prove infinity is to show that for every computable subclass of the given class we can effectively construct a new object from the class that does not lie in the subclass. We call such property of a class effective infinity.

The notion of effective infinity is similar to productiveness for a subset of the set of all natural numbers (Rogers [10]), but we intend to use the effective infinity for classes of computable enumerations. The notion of productiveness is useful for the study of r.e. sets, whereas we shall demonstrate here the usefulness of effective infinity for classes of computable enumerations.

The study of computable enumerations was established by Rogers [11] who defined the semilattice of computable enumerations for the families of all r.e. sets and all p.r. functions and also by Kolmogorov and Uspenskii [6]. Here we are interested mostly in subclasses of classes of minimal enumerations and classes of negative undecidable enumerations.

Friedberg [3] was the first to construct computable enumerations without repetitions (we call them here Friedberg enumerations) for the family of all r.e. sets and for the family of all p.r. functions. Then Pour-El [8] proved that there are at least two distinct Friedberg enumerations for each of these families. Malcev [7] introduced the notion of positive enumeration which implies minimality (as does the notion of the computable enumeration without repetition). Relying on the notion of positive enumeration Ershov [2] proved that there is a positive enumeration for each of the above families and such that it is not equivalent to any Friedberg enumeration of the respective families. His student Khutoretskii noticed that by using Ershov's method it is possible to construct a countably infinite number of positive pair-wise non-equivalent enumerations that are also not equivalent to any Friedberg enumeration. Also Khutoretskii [5] constructed minimal enumerations which are not positive.

Goncharov [4] raised the question about the size of the class of positive elements of a Roger's semilattice. In order to partially answer this question we do the following. Given a family of r.e. sets $S$ we consider certain classes of computable enumerations of $S$ and give sufficient conditions to be effectively infinite for these classes. The conditions are given in terms of restrictions on $S$, such as existence of an infinite subfamily of finite sets, existence of infinitely many supersets for any given finite member of the family and, say, existence of a Friedberg enumeration of the family. All or part of these conditions can be sometimes replaced by a requirement that the family has a Pour-El–Howard's height function. The classes that we consider are all the classes containing all Friedberg enumerations of $S$, all the classes containing all positive undecidable enumerations of $S$, all the classes containing all negative undecidable enumerations of $S$ and the class of all computable enumerations of $S$. We give several examples of families of r.e. sets satisfying the aforementioned requirements including the family of all r.e. sets. From these results the respective results of Pour-El [8], Pour-El and Howard [9], Ershov [2], and Khutoretskii [5] follow.
2. Approximating sequences of enumerations and their classes

An introduction to the theory of computable enumerations is given in Ershov's book [1]. For the sake of completeness we repeat some basic definitions which are borrowed from Ershov’s and Soare’s books with some insignificant changes.

We will use the following notational conventions. \( \omega \) is the set of all non-negative integers, \( \subseteq \) is the non-strict set-theoretical inclusion and \( \subset \) is the strict inclusion. Also, for all \( x, y \in \omega \), \( \langle x, y \rangle \) is the standard binary code of the pair \( (x, y) \). Finally, ‘s.t.’ abbreviates ‘such that’ and ‘iff’ abbreviates ‘if and only if’.

Usually, given non-empty sets \( A \subseteq B \), a function \( f : \omega \to B \) is called an enumeration of \( A \) if \( \text{Rng}(f) = A \). Since we can think of a function from \( \omega \) as an infinite sequence of objects, an enumeration of \( A \) can be described as an infinite sequence of elements of \( A \) which lists all \( A \). Later we will interchange the notions of a function from \( \omega \) and of an infinite sequence whenever it is convenient.

Let \( \mathcal{E} \) be the family of all r.e. sets. The class of all enumerations of some \( S \subseteq \mathcal{E} \) is designated \( H(S) \). We call \( Y \in H(S) \) a computable enumeration if there is some recursive \( h : \omega \to \omega \) s.t. for all \( X \), \( h(x) \) is equal to an r.e. index of \( Y(X) \).

Example 2.1. A family \( \{\emptyset\} \) has a unique enumeration which we designate \( \varepsilon \). It is easy to see that \( \varepsilon \) is computable.

The computability of an enumeration can be expressed in a different way. For a function \( \nu : \omega \to P(\omega) \) let \( G_{\nu} = \{ \langle x, y \rangle : y \in \nu(x) \land x \in \omega \} \).

**Proposition 2.1.** Let \( \nu : \omega \to P(\omega) \). Then \( \nu \) is a computable enumeration of some \( S \subseteq \mathcal{E} \) iff \( G_{\nu} = \{ \langle x, y \rangle : y \in \nu(x) \land x \in \omega \} \) is r.e.

For a computable enumeration \( \nu \) we define an index of \( \nu \) as an r.e. index of \( G_{\nu} \).

Let \( D_k \) be the finite set with the canonical finite set index \( k \). A sequence of finite sets \( \{A_x\}_{x \in \omega} \) is called a strong array if for some recursive function \( f \), \( A_x = D_f(x) \) for all \( x \in \omega \). A double-sequence \( \{A'_{x, y}\}_{x, y \in \omega} \) is called a strong double-array if \( \{B_{y}\}_{y \in \omega} \) is a strong array where for all \( x, y \in \omega \), \( A'_{x, y} = B_{f(x)} \). Strong triple-arrays and so forth are defined similarly.

A recursive approximating sequence\(^1\) of an r.e. set \( A \) is a strong array \( \{A_x\}_{x \in \omega} \) s.t. \( \forall x \in \omega [A_x \subseteq A_{x+1}] \) and \( A = \bigcup_{x \in \omega} A_x \). Similarly, for a sequence \( \nu \) of r.e. sets (i.e. \( \nu : \omega \to P(\omega) \)) a double-sequence \( \{\nu'(x)\}_{x, y \in \omega} \) of finite sets is called a recursive approximating sequence of \( \nu \) if \( \{\nu'(x)\}_{x, y \in \omega} \) is a strong double-array and for all \( x \in \omega \), \( \nu'(x) \) is a recursive approximating sequence of \( \nu(x) \).

**Proposition 2.2.** Let \( \nu : \omega \to P(\omega) \). Then \( \nu \) is a computable enumeration iff \( \nu \) admits a recursive approximating sequence.

\(^1\)In Soare’s book (see [12]) the same notion is called ‘recursive enumeration’.
Note that given an index of a computable enumeration \( \nu \), we can effectively produce a recursive approximating sequence of \( \nu \) and that the converse is also true.

Given computable enumerations \( \nu, \mu \) we say that \( \nu \text{ reduces to } \mu (\nu \equiv \mu) \) if there is a recursive function \( f : \omega \to \omega \) s.t. \( \nu = \mu \circ f \). We also say that \( \nu \) is equivalent to \( \mu (\nu = \mu) \) if \( \nu \equiv \mu \) and \( \mu \equiv \nu \). It is easy to see that \( \equiv \) is transitive but not anti-symmetric and that \( \leq \) induces a partial ordering on the equivalence classes of \( \equiv \). We call \( \nu \) a Friedberg enumeration (or without repetitions or injective) if \( \nu \) is injective, we call \( \nu \) positive if \( \eta_\nu = \{ (n, k) : \nu(n) = \nu(k) \& n, k \in \omega \} \) is r.e., we call \( \nu \) decidable if \( \eta_\nu \) is recursive and we call \( \nu \) negative if \( \Theta_\nu = \{ (n, k) : \nu(n) \neq \nu(k) \& n, k \in \omega \} \) is r.e. Also, we call \( \nu \in K \) minimal if \( \mu \equiv \nu \) implies \( \mu = \nu \) for any \( \mu \).

**Proposition 2.3.** If \( \nu \) is a positive enumeration then \( \nu \) is minimal.

**Proposition 2.4.** If \( S \) is an infinite family of r.e. sets and \( \nu \) is a computable enumeration of \( S \) then \( \nu \) is decidable iff \( \nu \) is equivalent to a Friedberg enumeration of \( S \).

A class of computable enumerations \( K \) is called a computable class of enumerations if there is an r.e. set \( H \) s.t. for any \( x \in H \), \( x \) is an index of some computable enumeration from \( K \) and, conversely, for each \( \nu \in K \) some index of \( \nu \) is in \( H \). In this case an r.e. index of \( H \) is called an index of \( K \). Just as for computable enumerations, the computability of \( K \) can be expressed in a different way.

Let \( K \) be a class of enumerations. We call a triple-sequence \( \{ \gamma_m(x) \}_{t, m \in \omega} \) of finite sets a recursive approximating sequence of \( K \) if \( \{ \gamma_m(x) \}_{t, m \in \omega} \) is a strong triple-array and if \( K = \emptyset \) then for every \( t, m, x \in \omega \), \( \gamma_m(x) = \emptyset \), otherwise for every \( \nu \in K \) there is some \( m \in \omega \) s.t. \( \{ \gamma_m(x) \}_{t, x \in \omega} \) is a recursive approximating sequence of \( \nu \) and conversely, for every \( m \in \omega \) there is some \( \nu \in K \) s.t. \( \{ \gamma_m(x) \}_{t, x \in \omega} \) is a recursive approximating sequence of \( \nu \). Let \( K \) be a class of enumerations and \( \{ \gamma_m(x) \}_{t, m \in \omega} \) be its recursive approximating sequence. Let us designate for all \( m, x \in \omega \), \( \gamma_m(x) = \bigcup_{t \in \omega} \gamma_m(x) \). It is easy to see that if \( K \) is not empty then \( \{ \gamma_m \}_{m \in \omega} \) is an enumeration of \( K \).

**Proposition 2.5.** Let \( K \) be a class of computable enumerations. Then \( K \) is computable iff \( K \) admits a recursive approximating sequence.

Note that given an index of a computable class of enumerations \( K \), we can effectively produce a recursive approximating sequence of \( K \). If we know that \( K \neq \emptyset \) then the converse is true. If, however, \( K \) may be empty then the converse is not true since both \( \emptyset \) and \( \{ \epsilon \} \) have the same unique recursive approximating sequence.
We will now introduce the following classes of enumerations.

\(K_{Fr}(S)\) = all Friedberg enumerations from \(H(S)\), 
\(K_{dec}(S)\) = all decidable enumerations from \(H(S)\), 
\(K_{min}(S)\) = all minimal enumerations from \(H(S)\), 
\(K_{pos}(S)\) = all positive enumerations from \(H(S)\), 
\(K_{neg}(S)\) = all negative enumerations from \(H(S)\), 
\(K_{pu}(S) = K_{pos}(S) - K_{dec}(S)\), i.e. all positive undecidable enumerations from \(H(S)\), 
\(K_{nu}(S) = K_{neg}(S) - K_{dec}(S)\), i.e. all negative undecidable enumerations from \(H(S)\), and 
\(K_{all}(S)\) = all computable enumerations from \(H(S)\). Note that if there is no confusion, we will sometimes write \(K_{Fr}, K_{dec}, K_{min}, K_{pos}, K_{neg}, K_{pu}, K_{nu}\) and \(K_{all}\).

3. Effective infinity of some classes of computable enumerations

We call a class \(K \subseteq H(S)\) effectively infinite if there is a p.r. function \(g\) s.t. for every \(x \in \omega\) if \(C\) is the computable class of enumerations with an index \(x\) and \(C \subseteq K\) then \(g(x) \downarrow\) and if \(v\) is the computable enumeration with an index \(g(x)\) then \(v \in K\) and \(\forall \xi \in C \left[\neg \xi \equiv v\right]\).

**Proposition 3.1.** If a class of computable enumerations \(K\) is effectively infinite then \(K\) is not computable and it contains a computable class of enumerations which is represented by an infinite sequence of pair-wise non-equivalent computable enumerations.

We will use the following notational conventions. For \(x, y \in \omega\) s.t. \(x \leq y\), \([x, y], (x, y], [x, y), (x, y)\) and \(x, y\) designate the intervals on \(\omega\) with \(x, y\) as the end points, where \(\lbrack, \rbrack\) mean that the end point is included and \(\lbrack, \rbrack\) mean that the end point is excluded. Also, for any set \(M \subseteq \omega\), \(M^c\) designates the complement of \(M\), i.e., \(M^c = \omega - M\).

**Theorem 1.** Assume that \(S \subseteq \mathcal{S}\) satisfies the following three properties.

(T1) There are infinitely many finite sets in \(S\).

(T2) For any finite set \(A \in S\) there are infinitely many distinct (not necessarily finite) sets \(B \in S\) s.t. \(A \subseteq B\).

(T3) There exists a Friedberg enumeration of \(S\).

Then any class \(K \subseteq K_{all}\) s.t. \(K_{Fr} \subseteq K\) is effectively infinite.

**Corollary 3.1.** For \(S = \mathcal{S}\), \(K_{Fr}, K_{dec}, K_{min}, K_{pos}, K_{neg}\) and \(K_{all}\) are effectively infinite.

**Proof.** (T1) and (T2) are obvious for \(\mathcal{S}\) and (T3) was shown by Friedberg (see [3]). □
Corollary 3.2 (Pour-El [8]). There exist two non-equivalent Friedberg enumerations of \( \mathbb{E} \).

The following example and proposition show that the assumption (T2) of Theorem 1 cannot be weakened to:

(T2') For any finite set \( A \in S \) there is some (not necessarily finite) set \( B \in S \) s.t. \( A \subset B \).

Example 3.1. Let \( S = \{ \omega \} \cup \{ \{ i \} : i \in \omega \} \). It is easy to see that (T1), (T2') and (T3) are true for \( S \) but (T2) is not true. Also, by Proposition 3.2, the conclusion of Theorem 1 fails for \( S \).

Proposition 3.2. All Friedberg enumerations of \( S \) from Example 3.1 are equivalent.

Proof. Let \( \nu, \mu \in K_{FR} \). To show \( \nu \sqsubseteq \mu \), we would like to find a recursive \( f \) s.t. \( \nu = \mu \circ f \). We can assume that we know such \( k, l \) that \( \nu(k) = \omega \) and \( \mu(l) = \omega \) and therefore we can define \( f(k) = l \). Now, suppose \( x \neq k \). Since \( \nu(x) \) consists of a unique element we can wait until \( G_\nu \) enumerates a pair of the form \( \langle x, x' \rangle \) and then wait until \( G_\nu \) enumerates a pair of the form \( \langle y, x' \rangle \). After that we define \( f(x) = y \). Note that we could do it because we know the number of distinct elements in \( \nu(x) \) and \( \mu(y) \) and that both sets are not empty in advance. \( \square \)

The next example and proposition illustrate that the assumption (T2) in Theorem 1 cannot be weakened to:

(T2'') For any distinct finite set \( A_1, \ldots, A_n \in S \) there are distinct (not necessarily finite) sets \( B_1, \ldots, B_n \in S \) s.t. for any \( i \in \{1, n\} \), \( A_i \subset B_i \).

Example 3.2. Define \( A_p = \{ p \} \) and \( B_p = \{ p^n : n \in \omega \} \) for all prime \( p \in \omega \) and let \( S = \{ A_p, B_p : p \) is prime and \( p \in \omega \} \). It is easy to see that (T1), (T2'') and (T3) are true for \( S \) but (T2) is not true. Also, by Proposition 3.3, the conclusion of Theorem 1 fails for \( S \).

Proposition 3.3. All Friedberg enumerations of \( S \) from Example 3.2 are equivalent.

Proof. Let \( \nu, \mu \in K_{FR} \). To show \( \nu \sqsubseteq \mu \), we would like to find a recursive \( f \) s.t. \( \nu = \mu \circ f \). Now, suppose \( x \in \omega \). Wait until \( G_\nu \) enumerates three pairs of the form \( \langle y, p \rangle \), \( \langle y', p^n \rangle \) and \( \langle y', p^m \rangle \) s.t. \( m \neq n \), \( y \neq y' \) and \( x \in \{ y, y' \} \) and \( G_\mu \) enumerates three pairs of the form \( \langle z, p \rangle \), \( \langle z', p^n \rangle \) and \( \langle z', p^m \rangle \) s.t. \( m \neq n \) and \( z \neq z' \). If \( x = y \) then define \( f(x) = z \), otherwise \( f(x) = z' \). \( \square \)

Theorem 2. Assume that \( S \subseteq \mathbb{E} \) satisfies (T1), (T2') and

(T3') There exists a positive enumeration of \( S \) (i.e., \( K_{pos} \neq \emptyset \)).

Then any class \( K \subseteq K_{all} \) s.t. \( K_{pu} \subseteq K \) is effectively infinite.
Corollary 3.3. For $S = \emptyset$, $K_{po}$ is effectively infinite.

Proof. $\emptyset$ satisfies (T3) which by Proposition 2.3 implies (T3'). □

Examples 3.1 and 3.2 show that the range of applications of Theorem 2 regarding the effective infinity of classes containing $K_{po}$ is strictly greater than that of Theorem 1. Indeed, according to Theorem 2, $K_{po}$ is effectively infinite in both examples whereas Theorem 1 is not applicable to either of them since (T2) is not satisfied. The next two examples further illustrate differences between applications but they exploit non-satisfaction of (T3). The examples show that there are such families of r.e. sets which admit an effectively infinite class of positive undecidable enumerations and do not admit any Friedberg enumeration at all.

Example 3.3. Let $M$ be any non-recursive r.e. set and let $S = \{\omega\} \cup \{(x): x \in M\}$. It is easy to see that $S$ satisfies (T1) and (T2') but not (T2).

Proposition 3.4. $S$ satisfies (T3') but does not satisfy (T3).

Proof. Let $S$, $M$ be as above. Define an enumeration of $S$ by $v(x) = \omega$ if $x \in M$ and $v(x) = \{x\}$ otherwise. Since $G_v = \{(x, y): x \in M, y \in \omega\} \cup \{(x, x): x \notin M\} = \{(x, y): x \in M, y \in \omega\} \cup \{(x, x): x \in \omega\}$ is r.e., $v$ is computable. It is easy to see that $\eta_v = \{(x, y): x \in M, y \in M\} \cup \{(x, x): x \in \omega\}$, so $v$ is positive. Therefore $S$ satisfies (T3').

We shall now show that $S$ does not have any Friedberg enumerations. Suppose that $\mu$ is a Friedberg enumeration of $S$. We can assume that we know such $k$ that $\mu(k) = \omega$. Consider the following effective procedure. List $G_v$. Whenever a pair $\langle x, y \rangle$ with $x \neq k$ is enumerated, output $y$. It is easy to see that this procedure effectively enumerates $M^c$ and so $M$ is recursive. Contradiction. □

Example 3.4. Let $M$ be any non-recursive r.e. set. For any $x \notin M$ and $y \in \omega$ let $A_{xy} = \{(x, n*y): n \in \omega\}$ and let $S = \{\omega\} \cup \{A_{xy}: x \notin M \text{ and } y \in \omega\}$. For any $\{(x, 0)\} \in S$ and any $y \in \omega$, $\{(x, 0)\} \subseteq A_{xy}$. It is easy to see that $S$ satisfies (T1) and (T2).

Proposition 3.5. $S$ satisfies (T3') but does not satisfy (T3).

Proof. Let $S$, $M$ be as above. Define an enumeration of $S$ by $v(\langle x, y \rangle) = \omega$ for every $x \in M$ and $y \in \omega$ and $v(\langle x, y \rangle) = A_{xy}$ for every $x \notin M$ and $y \in \omega$. Since $G_v = \{\langle(x, y), k\rangle: x \in M, y, k \in \omega\} \cup \{\langle(x, y), (x, n*y)\rangle: x \notin M, y, n \in \omega\} = \{\langle(x, y), k\rangle: x \in M, y, k \in \omega\} \cup \{\langle(x, y), (x, n*y)\rangle: x \in \omega, y, n \in \omega\}$ is r.e., $v$ is computable.
It is easy to see that $\eta_v = \{ (\langle x, y \rangle, \langle x', y' \rangle) : x, x' \in M, y, y' \in \omega \} \cup \{ \langle x, x \rangle : x \in \omega \}$, so $v$ is positive. Therefore $S$ satisfies (T3').

Just like for the previous example, it is easy to see that $S$ does not have any Friedberg enumerations. □

**Corollary 3.4** (Ershov [1] and Khutoretskii [5]). There is a countably infinite class of pair-wise non-comparable (in the sense of $\subseteq$) positive enumerations of $\mathcal{E}$ that are not equivalent to any Friedberg enumeration of $\mathcal{E}$.

**Proof.** By Corollary 3.3, $K_{nu}$ is effectively infinite for $\mathcal{E}$. By Proposition 3.1 there is an infinite sequence $C \subseteq K$ of pair-wise non-equivalent computable enumerations. By Proposition 2.3 any element of $K$ is a minimal enumeration and hence the elements of $C$ are pair-wise not comparable.

By Proposition 2.4 since all elements of $C$ are undecidable, they are not equivalent to any Friedberg enumeration of $\mathcal{E}$. □

**Theorem 3.** Assume that $S \subseteq \mathcal{E}$ satisfies (T1), (T2) and

(T3') There exists a negative enumeration of $S$ (i.e., $K_{neg} \neq \emptyset$).

Then any class $K \subseteq K_{all}$ s.t. $K_{nu} \subseteq K$ is effectively infinite.

**Corollary 3.5.** For $S = \mathcal{E}$, $K_{nu}$ is effectively infinite.

Example 3.1 and the following proposition show that the assumption (T2) of Theorem 3 cannot be weakened to (T2').

**Proposition 3.6.** All negative enumerations of $S$ from Example 3.1 are equivalent.

**Proof.** It is enough to show that every negative enumeration of $S$ is decidable. Suppose $v$ is negative. Then $\theta_v$ is r.e. We would like to show that $\theta_v$ is recursive. It is enough to show that we can decide for every $i$ what the value of $v(i)$ is. The rest would easily follow.

Take some $i \in \omega$. Enumerate $G_v$ and $\theta_v$. Then if $v(i) = \omega$ then $G_v$ will enumerate $\langle i, k \rangle$ and $\langle i, l \rangle$ with $k \neq l$. Otherwise (i.e., if $v(i)$ is a one-element set) $G_v$ will enumerate $\langle i, k \rangle$, $\langle j, l \rangle$ and $\langle j, m \rangle$ with $i \neq j$ and $l \neq m$ and $\theta_v$ will enumerate $\langle i, j \rangle$ in which case we would conclude that $v(i) = \{k\}$. □

Example 3.2 and the following proposition show that the assumption (T2) of Theorem 3 cannot be weakened to (T2').

**Proposition 3.7.** All negative enumerations of $S$ from Example 3.2 are equivalent.

**Proof.** The idea of the proof is the same as above. □
The following definition will be needed to state and prove a strengthened form of the theorem of Pour-El and Howard about existence of Friedberg enumerations for certain families of r.e. sets.

**Definition** (Pour-El and Howard [9]). We say that a family $S$ of subsets of $\omega$ has a *height function* $h$ if the domain of $h$ is the family $S'$ of all finite subsets of members of $S$, the range of $h$ lies in $\omega$ and the following conditions hold:

1. $h$ is monotonic, i.e., whenever $A \subseteq B$, $h(A) \leq h(B)$ for all $A, B \in S'$.
2. $h$ satisfies the following ascending chain condition. Given any ascending sequence $A_0 \subseteq A_1 \subseteq \cdots \subseteq A_n \subseteq \cdots$ of finite subsets of a fixed member of $S$, the associated sequence of heights $h(A_0) \leq h(A_1) \leq \cdots \leq h(A_n) \leq \cdots$ eventually becomes constant.
3. For every finite $A \in S'$ there is $B \in S'$ s.t. $A \subseteq B$ and $h(A) \neq h(B)$.

The conclusion of Theorem 1 can be obtained if the first two conditions for a family $S$ of r.e. sets are replaced by a condition that $S$ has a height function. The following theorem is a strengthened form of the Theorem 1 from [9].

**Theorem 4.** Suppose that:

(H) $S$ has a height function.

(C) $S$ has a computable enumeration.

Then any class $K \subseteq K_{\text{all}}$ s.t. $K_{Fr} \subseteq K$ is effectively infinite.

The following corollary gives more examples of computable families and some of their effectively infinite classes of enumerations.

**Corollary 3.7.** The below indicated classes $K \subseteq K_{\text{all}}(S)$ are effectively infinite for the following computable families $S$:

1. $S$ is a family of finite sets s.t. for every $A \in S$ there is $B \in S$ s.t. $A \subseteq B$ and $K$ is any class s.t. $K_{Fr} \subseteq K$ or $K_{nu} \subseteq K$ or $K_{pa} \subseteq K$.
2. $S$ is an infinite family of finite sets which is linearly ordered under inclusion and $K$ is any class s.t. $K_{Fr} \subseteq K$ or $K_{nu} \subseteq K$ or $K_{pa} \subseteq K$.
3. $S$ is such that for every $x \in \omega$, $S$ includes a set which contains the segment $[0, x)$ and $\omega \notin S$ and $K$ is any class s.t. $K_{Fr} \subseteq K$.
4. For every two finite sets $A, B$ contained in some members of $S$, the set $A \cup B$ is also contained in some member of $S$ and $\bigcup S \notin S$ and $K$ is any class s.t. $K_{Fr} \subseteq K$.
5. $S = \emptyset - \{\omega\}$ and $K$ is any class s.t. $K_{Fr} \subseteq K$ or $K_{nu} \subseteq K$ or $K_{pa} \subseteq K$.

**Proof.** For each of cases (1)–(4) the height functions are given in [9] and hence effective infinity of $K \subseteq K_{Fr}$ follows from Theorem 4. The same conclusion for case (5) follows from that of the case (3). Then for families (1), (2), (5) effective infinity of the remaining classes follows from the Theorems 2 and 3. □
4. Friedberg enumerations: a proof of Theorem 1

We will use the following notational conventions. If \( h : A \rightarrow B \) is a function and \( U \subseteq A \) then we abuse notation by designating \( h(U) = \{h(x) : x \in U\} \). Also, for all \( x, y, n, t \in \omega, f_n \) is the p.r. function represented by the Turing program with Gödel number \( n \) and \( f_n(x) = y \) if \( x, y, n < t \) and \( y \) is the output obtained in less than \( t \) steps by the Turing program with Gödel number \( n \).

Assume that we are given a recursive approximating sequence \( \{\mu'_{x}(x)\}_{x \in \omega} \) of a computable enumeration \( \mu \in K_{Fr} \) and a recursive approximating sequence \( \{\gamma'_{m}(x)\}_{m, x \in \omega} \) of a computable class \( C \subseteq K \). We would like to construct a computable enumeration \( \nu \in K \) s.t. if \( C \) is not empty then \( \gamma'_{m} = \nu \) for all \( m \in \omega \).

4.1. Insuring that \( \nu \in K \)

First we have to guarantee that \( \nu \) is a computable enumeration. To do that it is enough to construct its recursive approximating sequence \( \{\nu'(x)\}_{x \in \omega} \). Intuitively, to construct a recursive function defining the corresponding strong array it is enough to build for every \( t, x \in \omega \) a terminating effective procedure which creates finite sets \( \nu'(x) \) in such a way that for any \( t, x \in \omega \), \( \nu'(x) \subseteq \nu'^{+1}(x) \).

While constructing \( \nu \) we shall ensure that there is a (not necessarily recursive) function \( \varphi : \omega \rightarrow \omega \) s.t. \( \varphi \) is bijective and \( \nu = \mu \circ \varphi \). This will guarantee that \( \nu \in K_{Fr} \) and hence \( \nu \in K \).

We intend to build \( \varphi \) as a point-wise limit of a sequence of recursive functions \( \{\varphi'_{t}\}_{t \in \omega} \). Since we are in the domain of the natural numbers, it is easy to see that a sequence of functions \( \varphi' : \omega \rightarrow \omega \) point-wise converges for \( t \rightarrow \infty \) to a function \( \varphi \) (or stabilizes into a function \( \varphi \)) iff for any \( x \in \omega \) there is some \( t' \) that for any \( t \geq t' \), \( \varphi'(x) = \varphi''(x) = \varphi(x) \).

The following lemma gives sufficient conditions for stabilization of \( \varphi' \), for injectiveness of \( \varphi \) and for surjectiveness of \( \varphi \).

**Lemma 4.1.** Suppose we have a sequence of finite sets \( \text{ForbZone}^{t} \subseteq \omega \) (which stands for 'forbidden zone') and a sequence of integers \( a^{t} \) where \( t \in \omega \). Define the following five properties.

1. \( a^{t} \rightarrow \infty \) with \( t \rightarrow \infty \).
2. \( [0, a^{t}] \subseteq \text{ForbZone}^{t} \) for all \( t \).
3. \( \varphi'^{t+1} \upharpoonright \text{ForbZone}^{t+1} = \varphi'^{t} \upharpoonright \text{ForbZone}^{t+1} \) for all \( t \).
4. \( \varphi'^{t} \upharpoonright [0, a^{t}] \) is injective for all \( t \).
5. \( \varphi'^{t} \upharpoonright (-1)([0, a^{t}]) \subseteq \text{ForbZone}^{t} \) for all \( t \) and \( [0, a^{t}] \subseteq \text{Rng}(\varphi') \).

Then the following is true.

(a) \( (1), (2), (3) \Rightarrow \text{sequence } \varphi' \text{ converges with } t \rightarrow \infty \).
(b) \( (1), (2), (3), (4) \Rightarrow \text{the limit of } \varphi' \text{ is injective} \).
(c) \( (1), (2), (3), (4), (5) \Rightarrow \text{the limit of } \varphi' \text{ is surjective}^2 \).

\(^2\) We separate the conditions for injectiveness and surjectiveness since Lemma 4.1 is also used in Theorems 2 and 3 where injectiveness is not required.
The following Lemma shows how to construct a recursive approximating sequence \( \{v'(x)\}_{t,x \in \omega} \) for a computable enumeration \( v \) and to achieve \( v = \mu \circ \varphi \).

**Lemma 4.2.** If

1. \( \{v'(x)\}_{t,x \in \omega} \) is a sequence of finite sets s.t. \( v'(x) \subseteq v^{t+1}(x) \) for all \( t \) and \( x \in [0, t) \) and \( v'(x) = \emptyset \) for all \( t \) and \( x \geq t \);
2. \( \{\varphi'(x)\}_{t,x \in \omega} \) is a sequence of functions s.t. \( \psi \) is recursive where \( \forall t, x \psi(t, x) = \varphi'(x) \);
3. \( e: \omega \to \omega \) is a recursive function s.t. \( e(t) \to \infty \) with \( t \to \infty \);
4. \( v'(x) = \mu^{e(t)}(\varphi'(x)) \) for all \( t \) and \( x \in [0, t) \),

then \( v \) is a computable enumeration and \( \{v'(x)\}_{t,x \in \omega} \) is a recursive approximating sequence of \( v \), where for all \( x \in \omega \), \( v(x) = \bigcup_{t \in \omega} v'(x) \). Moreover, if

5. the sequence \( \varphi'(x) \) stabilizes with \( t \to \infty \) into a function \( \varphi \), then \( v = \mu \circ \varphi \).

In the algorithm below we are going to implement the conditions of Lemmas 4.1 and 4.2.

### 4.2. Insuring that \( \neg \gamma_m = v \) for all \( m \in \omega \)

If \( \gamma_m = v \) then \( \gamma_m \subseteq v \) and hence for some \( n \in \omega \) s.t. \( f_n \) is recursive we would have \( \gamma_m = v \circ f_n \). Therefore we will achieve \( \neg \gamma_m = v \) if we construct \( v \) in such a way that for every \( n \in \omega \) if \( f_n \) is recursive then there is \( s \in \omega \) s.t. \( \gamma_m(s) \) is finite and \( \gamma_m(s) \subseteq v(\varphi(n)) \).

Note that at each stage \( t \) we would like to fulfil the requirements (3) from Lemma 4.1 and (1), (4) from Lemma 4.2, and that we don't know for which \( y \) \( \gamma_m(y) \) is finite. So we intend to strictly enlarge the sets of the form \( v'(f_n(y)) \) whenever we can without violating the aforementioned requirements, trying to exceed possibly finite \( \gamma_m(y) \). To do that we will define \( q^{t+1} \) and \( e(t+1) \) in such a way that for all \( x \in \text{Rng}(f_n^{t+1}) \cdot \text{ForbZone}^{t+1} \) we would have \( v'(x) \subseteq \mu^{e(t+1)}(q^{t+1}(x)) \). By (4) from Lemma 4.2 and since \( \text{Rng}(f_n^{t+1}) \subseteq [0, t] \), it would mean that for all \( y \) s.t. \( f_n^{t+1}(y) \in [0, t] \cdot \text{ForbZone}^{t+1} \) we would have \( v'(f_n(y)) \subseteq v^{t+1}(f_n(y)) \). Later we will show that this will lead to the existence of some \( s \) as above.

### 4.3. The algorithm

We shall use the following notational conventions. If \( \psi \) is a function which is being defined and exp is some expression then assignment \( \psi(x) := \text{exp} \) means "define \( \psi(x) \) to be as \( \text{exp} \)". The algorithm represents an inductive definition on stages \( t \). The procedure for stage \( t + 1 \) contain several regions of the form

\[
\begin{align*}
(Rk) & \quad / * \cdots * / \\
& \quad \text{begin} \\
& \quad \cdots \\
& \quad \text{end}
\end{align*}
\]
Here the purpose of the comments \(/ * \cdots */\) is to give a brief description of the actions inside the brackets \texttt{begin} \cdots \texttt{end}.

**Definition.** 1. \(FZ(x, t) = [0, x) \cup (q')^{-1}([0, x))\) for all \(x, t \in \omega\).

2. A pair \((m, n)\) requires attention at \(t+1\) if
   \begin{enumerate}
   
   \item \(m < t + 1\) and \(n < t + 1\);
   
   \item \(0, r'(m, n)) \subseteq \text{Dom}(f^{t+1}_n)\) and for all \(x < r'(m, n), y^{t+1}_n(x) \cap [0, r'(m, n)) = y'(f^{t+1}_n(x)) \cap [0, r'(m, n))\). Here \(r'(m, n)\) is the ‘restraint’ for \((m, n)\) at stage \(t\). It is defined recursively and serves to restrict the domains of functions to \([0, r'(m, n))\);
   
   \item \(\text{Rng}(f^{t+1}_n) - FZ(\max(m, n), t) \neq \emptyset\).
   
   \end{enumerate}

3. A pair \((m, n)\) is active at \(t+1\) if \((m, n)\) is the \(\leq\)-least of all pairs which requires attention at \(t+1\), where \((m, n) \leq (m', n')\) if \((m, n) \leq (m', n')\).

**Construction**

\textbf{Stage 0.} \(\psi^0 := \text{Id}_{\omega}, \forall x \in \omega \ [\psi^0(x) := \emptyset], a' := 0, \text{ForbZone}^0 := \emptyset, e(t) := 0\) and \(\forall m, n \in \omega \ [r^0(m, n) := 0]\).

\textbf{Stage } \(t+1\).

\texttt{begin}

if there is an active pair \((m, n)\) at \(t+1\)

\texttt{then}

\texttt{begin}

\texttt{t}^{++1} := \max(m, n) \text{ and } \text{ForbZone}^{t+1} := FZ(\max(m, n), t);\\

\textbf{(R1) } /* \text{Find } \psi \text{ and } t' \geq e(t) \text{ s.t. } \psi[0, t] \cup \text{ForbZone}^{t+1} \text{ is injective and s.t. for all } x \in \omega \text{ if } x \in \text{Rng}(f^{t+1}_n)-\text{ForbZone}^{t+1} \text{ then } y'(x) \subseteq \mu'(\psi'(x)) \text{ and otherwise } \psi(x) = \psi'(x) */\\

\texttt{begin}

Let \(\{x_1, \ldots, x_q\} = \text{Rng}(f^{t+1}_n)-\text{ForbZone}^{t+1}\), where \(x_1, \ldots, x_q\) are distinct; Find the least \(t' \geq e(t)\) s.t. there are distinct \(z_1, \ldots, z_q\) s.t. for every \(i \in [1, q]\) at least one of the following holds:

\begin{enumerate}
   
   \item \(x_i = z_i \text{ and } y'(x_i) \subseteq \mu'(\psi'(x_i))\) or
   
   \item \(z_i \in [0, t') - \psi'([0, t) \cup \text{ForbZone}^{t+1}) \text{ and } y'(x_i) \subseteq \mu'(z_i)\);
   
   \end{enumerate}

Define \(\psi\) as follows. If for some \(x \in \omega\) there is \(i \in [1, q]\) s.t. \(x = x_i\) and \(x_i \neq z_i\) then \(\psi(x) := z_i\), otherwise \(\psi(x) := \psi'(x)\);

\texttt{end}

\textbf{(R2) } /* \text{Insure that } \varphi^{t+1}: \omega \rightarrow \omega \text{ is bijective } */\\

\texttt{begin}

Define \(\varphi^{t+1}|([t+1, t')-\text{ForbZone}^{t+1})\) as the unique monotonically increasing bijection between \([t+1, t')-\text{ForbZone}^{t+1}\) and \([0, t') - \psi([0, t] \cup \text{ForbZone}^{t+1})\); For \(x \in [0, t] \cup \text{ForbZone}^{t+1}\), \(\varphi^{t+1}(x) := \psi(x)\) and for \(x \geq t', \varphi^{t+1}(x) := x\);

\texttt{end}

\(e(t+1) := \max(t + 1, t'), r^{t+1}(m, n) := r'(m, n) + 1\) and \(\forall m', n' \in \omega \ [(m', n') \neq (m, n) \leftrightarrow r^{t+1}(m, n) := r'(m, n)]\).
Some effectively infinite classes of enumerations

end
else
begin
  \( q_{t+1} := q', a_{t+1} := a', \) \( \text{ForbZone}_{t+1} := \text{ForbZone}' \), \( e(t + 1) := \max(t + 1, e(t)) \) and \( r_{t+1} := r' \);
end

\( \forall x \in [0, t] \ [v^{t+1}(x) := \mu^{t+1}(q^{t+1}(x))] \) and \( \forall x \in [t, \infty) \ [v^{t+1}(x) := \emptyset] \)
end of Stage \( t + 1 \).

Lemma 4.3. If \( q' \) is bijective, \( (q'([0, t])) \cup (q')^{-1}([0, t])) \subseteq [0, e(t)) \), \( v'(x) = \mu^{t+1}(q'(x)) \) for any \( x \in [0, t) \) and \( v'(t) = \emptyset \) then the stage \( t + 1 \) terminates.

(Note that here we are using only (T2).)

Proof. The finite nature of the algorithm is disguised by the fact that at the stage \( t + 1 \) we are defining a total function \( q^{t+1} \), but nevertheless the computations are finite since \( q^{t+1} \) differs from \( q' \) only in a finite number of points. The same is true for all other sequences of functions.

It is easy to see that (R2) terminates. We will show that (R1) terminates. Let \( j \in [1, q] \). Then \( v'(x_j) = \mu^{t+1}(\psi(x_j)) \) or \( v'(x_j) = \emptyset \). If \( \mu(\psi(x_j)) \) is infinite then since \( \{\mu'(x)\}_{t \leq t_0} \) is a recursive approximating sequence of \( \mu \) there is some \( t_0 \geq e(t) \) s.t. \( NE1 \) is true for \( i = j \) and all \( t'' > t_0 \). If \( \mu(\psi(x_j)) \) is finite then by (T2) there are infinitely many such \( z \) s.t. \( \mu'(x_j) \subseteq \mu(z) \). Since \( \psi([0, t]) \) is finite it is obvious that (NE2) will occur and that we can attain \( z_i \neq z_j \) for \( i \neq j \). \( \square \)

Corollary 4.1. For any \( t \in \omega \), \( q' \) is bijective, \( (q'([0, t])) \cup (q')^{-1}([0, t])) \subseteq [0, e(t)) \), \( v'(x) = \mu^{t+1}(q'(x)) \) for any \( x \in [0, t) \) and \( v'(t) = \emptyset \).

Corollary 4.2. Conditions (1)–(4) of Lemma 4.2 are satisfied.

Let \( v \) be such a sequence of sets that for all \( x \in \omega \), \( v(x) = \bigcup_{t \leq t_0} v'(x) \).

Corollary 4.3. \( v \) is a computable enumeration and \( \{v'(x)\}_{t, x \in \omega} \) is a recursive approximating sequence of \( v \).

Corollary 4.4. For all \( t \in \omega \) and \( x \in \text{Rng}(f^{t+1}) \cdot \text{ForbZone}^{t+1} \) we have \( v'(x) \subseteq v^{t+1}(x) \).

Lemma 4.4. Each pair \((m, n)\) could be active only at finitely many stages.\(^3\)

(Note that here we are using only (T1), (1) and (4) from Lemma 4.2 and Corollary 4.4.)

\(^3\) It is not necessary to use the fact that \( q' \) is injective, which makes it possible to modify Lemma 4.4 for Theorem 3.
Proof. Suppose \((m, n)\) is the \(\leq\)-minimal pair which is active infinitely often. Then by the algorithm \(r'(m, n) \rightarrow \infty\) with \(t \rightarrow \infty\). From this, \((A2)\) and Corollary 4.1 (i.e., \((1)\) from Lemma 4.2) it follows that \(\gamma_m = \nu \circ f_n\).

If \(C\) is empty then \(\gamma_m = \varepsilon\) (see Example 2.1 and the definition of a recursive approximating sequence of a class of enumerations). Also, it is easy to see that if \((m, n)\) was active at least once then for some \(y \in \omega\), \(\nu \circ f_n(y) \neq \emptyset\), which gives a contradiction. Now we can assume that \(C\) is not empty and hence \(\gamma_m \in K\).

By Corollary 1 we have \((4)\) from Lemma 4.2. From this and the definition of \(FZ(x, t)\) it follows that there are no more then \(2 \max(m, n)\) distinct elements in \(\nu'(FZ(\max(m, n), t) \cap [0, t))\) for all \(t\).

Since \(\gamma_m \in K\) by \((T1)\) there are distinct \(y_1, \ldots, y_p \in \text{Dom}(f_n)\) with \(p > 2 \max(m, n)\) and s.t. \(\gamma_m(y_1), \ldots, \gamma_m(y_p)\) are finite and distinct. Since \(\gamma_m = \nu \circ f_n\), we have that \(f_n(y_1), \ldots, f_n(y_p)\) are also distinct.

Then there is \(t\) s.t. \((m, n)\) is active at \(t + 1\), \(y_1, \ldots, y_p \in \text{Dom}(f_{n+1})\) and for all \(i \in [1, p]\), \(\nu'(f_n(y_i)) = \nu(f_n(y_i))\) and \(f_n(y_i) \in [0, t)\). Then \(FZ(\max(m, n), t) = \text{ForbZone}'^{n+1}\). It is easy to see that there is some \(i \in [1, p]\) s.t. \(f_n(y_i) \in \text{Rng}(f_{n+1})\)-\text{ForbZone}'^{n+1}. By Corollary 4.4, \(\nu'(f_n(y_i)) \subseteq \nu^{n+1}(f_n(y_i))\) and hence \(\nu(f_n(y_i)) \subseteq \nu^{n+1}(f_n(y_i))\). Contradiction. \(\square\)

Corollary 4.5. Conditions \((2)-(5)\) of Lemma 4.1 are satisfied.

Corollary 4.6. For all \(t \in \omega\) if \(\max(m, n) \leq a^{t+1}\) and \([0, \max(m, n)) \subseteq \text{Rng}(q')\) then \(FZ(\max(m, n), t) = FZ(\max(m, n), t + 1)\).

Lemma 4.5. For all \(m, n \in \omega\) if \(\text{Rng}(f_n)\) is infinite then \((A3)\) is satisfied infinitely often.

(Note that here we are using only Lemma 4.4, \((3)\) and \((5)\) from Lemma 4.1 and Corollary 4.6.)

Proof. Let \((m, n)\) be such that \(\text{Rng}(f_n)\) is infinite and \(t\) be such that \((A3)\) is not satisfied at all \(t' \geq t\). By Lemma 4.4 we can assume without loss of generality that for any \(t' > t\) and all \(m', n'\) s.t. \(m' < m\) and \(n' < n\), \((m', n')\) is not active at \(t'\). If no pair is active at any \(t' > t\) then for any \(t' \geq t\), \(FZ(\max(m, n), t') = FZ(\max(m, n), t)\). If some pair is active at some \(t' \geq t\) we can assume without loss of generality that it is active at \(t\). Then by Corollary 4.5 (i.e., by \((5)\) from Lemma 4.1) and Corollary 4.6 we again have that for any \(t' \geq t\), \(FZ(\max(m, n), t') = FZ(\max(m, n), t)\). If \(t' \geq t\) is s.t. \(\text{Rng}(f_{n+1}) - FZ(\max(m, n), t) \neq \emptyset\) then \((A3)\) is satisfied at \(t'\). Contradiction. \(\square\)

Corollary 4.7. The requirement \((1)\) from Lemma 4.1 holds.

Corollary 4.8. \(\{q_i\} \in \omega\) converges. Moreover, if \(\varphi = \lim_{r \to \infty} q_i\) then \(\varphi\) is bijective and \(\nu = \mu \circ \varphi\).
Corollary 4.9. \( \forall \in K \).

Lemma 4.6. If \( C \) is not empty then for all \( m \in \omega \), \( \neg \gamma_m = \forall \).

(Note that here we are using only (T1) and Lemma 4.5.)

Proof. Suppose \( \gamma_m = \forall \). Then for some recursive \( f_n \) we would have \( \gamma_m = \forall \circ f_n \).

Since \( C \) is not empty, \( \gamma_m \in K \) and hence by (T1) \( \text{Rng}(f_n) \) is infinite. So by Lemma 4.5 (A3) is satisfied infinitely often. By Lemma 4.4 there is a stage \( t' > \max(m, n) \)

s.t. no pair \((m', n') \leq (m, n)\) is active at any \( t \geq t' \). Since \( \gamma_m = \forall \circ f_n \), it is easy to see that there is \( t'' \geq t' \) s.t. (A1) and (A2) are true at all \( t \geq t'' \). So if \( t \geq t'' \) is s.t. (A3) is satisfied at \( t \) then \((m, n)\) is active at \( t \). Contradiction. \( \square \)

5. Positive undecidable enumerations: a proof of Theorem 2

We will use the following notational conventions. \( \chi_k(x, y) = f_k((x, y)) \) if \( x, y < t \) and the right-hand side is defined. Also, if \( R \) is a binary relation on \( \omega \) and \( x \in \omega \) then \( R(x) = \{ y : xRy \} \) and if \( U \subseteq \omega \) then \( R(U) = \{ y : \exists x \in U \mid xRy \} \). Finally, \( \text{Equiv}(R) \) called the equivalence-closure of \( R \) is the smallest equivalence containing \( R \).

Assume that we are given a recursive approximating sequence \( \{ \mu'(x) \}_{t \in \omega} \) of a computable enumeration \( \mu \in K_{\text{pos}} \), a recursive approximating sequence \( \{ \eta'_t \}_{t \in \omega} \) of \( \eta_\mu \) and a recursive approximating sequence \( \{ \gamma'_m(x) \}_{t, m \in \omega} \) of a computable class \( C \subseteq K \). We would like to construct a computable enumeration \( \nu \in K \) s.t. if \( C \) is not empty then \( \neg \gamma_m = \forall \) for all \( m \in \omega \).

5.1. Insuring that \( \nu \in K \)

We shall guarantee that \( \nu \) is a computable enumeration in the same way as in Section 4.

Further, we shall build \( \nu \) in such a way that there is \( \varphi: \omega \rightarrow \omega \) s.t. \( \varphi \) is surjective and \( \nu = \mu \circ \varphi \). This will guarantee that \( \nu \in H(S) \). To guarantee \( \nu \in K \) we will have to insure that \( \nu \) is positive and not decidable. This is the same as insuring that \( \eta_\nu \) is r.e. but not recursive.

We shall guarantee that \( \eta_\nu \) is r.e. as follows. Assume that \( \nu = \mu \circ \varphi \) and \( \varphi = \lim_{t \rightarrow t'} \varphi' \) where \( \{ \varphi'(x) \}_{t \in \omega} \) is a sequence of recursive functions. Let us define a sequence \( \{ \eta_{t'} \}_{t \in \omega} \) of finite sets by \( \eta_{t'} = \{ (x, y) : x, y \in [0, t) \} \). It is easy to see that this definition guarantees that \( \{ \eta_{t'} \}_{t \in \omega} \) is a strong array and that therefore \( \bigcup_{t \in \omega} \eta_{t'} \) is r.e. The next step is to insure that \( \bigcup_{t \in \omega} \eta_{t'} = \eta_\nu \). Since \( \{ \eta_{t'} \}_{t \in \omega} \) is a recursive approximating sequence of \( \eta_\mu \) it is easy to show that \( \eta_{t'} \subseteq \bigcup_{t \in \omega} \eta_{t'} \). All of the above together with \( \forall t \in \omega \) \( \eta_{t'} \subseteq \eta_{t'+1} \) would guarantee us that \( \bigcup_{t \in \omega} \eta_{t'} = \eta_\nu \).

Now we will show how to insure that for all \( t \in \omega \), \( \eta_{t'} \subseteq \eta_{t'+1} \). Consider the family of equivalence classes which are induced on \( \omega \) by \( \text{Equiv}(\eta_{t'}) \). For \( x \in \omega \) the
class that contains \( x \) is designated \( \text{Cl}'(x) \). We shall use these classes in the construction by defining \( \{ \eta^t \}_{t \in \omega} \) in such a way that for every \( t, x \in \omega \), either \( \eta^{t+1} \) is identical with \( \eta^t \) over \( \text{Cl}'(x) \) or \( \eta^{t+1} \) is constant over \( \text{Cl}'(x) \). It is easy to check that this guarantees \( \eta^t \subseteq \eta^{t+1} \). Moreover, this property of \( \eta^{t+1} \) guarantees \( \eta^t \subseteq \eta^{t+1} \) and \( \bigcup_{t \in \omega} \eta^t \subseteq \eta_n \) without presupposing that \( \forall = \mu \circ \forall \) or \( \forall = \lim_{t \to \infty} \forall^t \) but with only presupposing \( (1) \) and \( (4) \) from Lemma 4.2. In the verification of the algorithm the inclusion \( \bigcup_{t \in \omega} \eta^t \subseteq \eta_n \) is used to show that \( \forall = \lim_{t \to \infty} \forall^t \).

We insure that \( \eta_n \) is not recursive as follows. If \( \eta_n \) were recursive then the characteristic function of \( \eta_n \) would be a two-placed recursive function. So we will try to insure that for any \( k \) the \( k \)th two-placed p.r. function \( \chi_k \) is \textit{not} a characteristic function of \( \eta_n \). To do that it is enough to construct \( \eta_n \) and \( v \) in such a way that for any \( k \) there is a pair \( \langle x, y \rangle \) s.t. \( \langle x, y \rangle \in \eta_n \) and \( \chi_k(x, y) = 0 \) or \( \langle x, y \rangle \notin \eta_n \) and \( \chi_k(x, y) = 1 \). The present algorithm explicitly takes care only of the first clause of this disjunction. The opportunity to do so lies in the fact that a characteristic function of \( \eta_n \) has two properties which we intend to violate. The first is that for all pairs \( \langle x, y \rangle \), \( \chi_k(x, y) = 0 \) implies \( \langle x, y \rangle \notin \eta_n \). Since for all \( t, \eta^t \subseteq \eta_n \), this permits us to deal with \( \chi_k \) only on those stages \( t + 1 \) where

For all \( x, y < t + 1 \) s.t. \( \langle x, y \rangle \in \text{Dom}(\chi^{t+1}_k) \), \( \chi^{t+1}_k(x, y) \in \{0, 1\} \) and if \( \chi^{t+1}_k(x, y) = 0 \) then \( \langle x, y \rangle \notin \text{Equiv}(\eta_n) \).

The second property is that there are infinitely many pairs \( x, y \) s.t. \( \chi_k(x, y) = 0 \) and \( v(x) = v(y) \) (this immediately follows from (T1) and (T2)). This permits us to deal with \( \chi_k \) only on those stages \( t + 1 \) when it satisfies

There is \( t' \) s.t. for all \( t \geq t' \) there are \( x, z \in [0, t] \) s.t. \( x \notin \text{FZ}'(k, t) \), \( \chi_k(x, z) = 0 \) and \( \forall y \in \text{Cl}'(x) [v'(y) \subseteq v'(z)] \).

Then we make \( \langle x, z \rangle \in \eta^{t+1}_n \) by putting for all \( y \in \text{Cl}'(x) \), \( \eta^{t+1}_n(y) := \eta^{t+1}_n(z) \), and for all \( y \notin \text{Cl}'(x) \), \( \eta^{t+1}_n(y) := \eta^{t+1}_n(y) \). The notion of ‘requires attention’ for the number \( k \) reflects the above two properties as (AN2) and (AN3) below.

5.2. The algorithm

We will use some notions defined in Section 4.3 except those which we redefine here.

**Definition.** 1. For all \( x \in \omega \), \( \text{Cl}'(x) = \{ y : \langle x, y \rangle \in \text{Equiv}(\eta_n) \} \) is called the equivalence class of \( x \) at stage \( t \).

2. \( \text{FZ}'(x, t) = \text{Equiv}(\eta_n)(\text{FZ}(x, t)) \) for all \( x, t \in \omega \).

3. A pair \( (m, n) \) requires attention at \( t + 1 \) if (A1), (A2) and (A3') \( \text{Rng}(f^{t+1}_n) - \text{FZ}'(\max(m, n), t) \neq \emptyset \).

4. A number \( k \) requires attention at \( t + 1 \) if
   - (AN1) \( k < t + 1 \);
   - (AN2) for all \( x, y < t + 1 \) s.t. \( (x, y) \in \text{Dom}(\chi^{t+1}_k) \), \( \chi^{t+1}_k(x, y) \in \{0, 1\} \) and if \( \chi^{t+1}_k(x, y) = 0 \) then \( \langle x, y \rangle \notin \text{Equiv}(\eta_n) \);
(AN3) there are $x, z \in [0, t]$ s.t. $x \notin FZ'(k, t), \chi^t_{x}(x, z) = 0$ and $\forall y \in Cl'(x) [v'(y) \subseteq v'(z)]$.

5. A pair $(m, n)$ is active at $t + 1$ if $t + 1$ is odd and $(m, n)$ is the $<_{-}$-least of all pairs which requires attention at $t + 1$, where $(m, n) \leq (m', n')$ if $(m, n) \leq (m', n')$.

6. A number $k$ is active at $t + 1$ if $t + 1$ is even and $k$ is the least of all numbers which requires attention at $t + 1$.

Construction

Stage $0$. $q^0 := Id_v, \forall x \in \omega [v^0(x) := 0], \eta^0 := \emptyset, \alpha^t := 0, ForbZone' := \emptyset, e(t) := 0 \text{ and } \forall m, n \in \omega [v^0(m, n) := 0]$.

Stage $t + 1$.

begin
if there is an active pair $(m, n)$ at $t + 1$
then
begin
$a'^{t+1} := \max(m, n)$ and $ForbZone'^{t+1} := FZ'(\max(m, n), t)$;
Define $q'^{t+1}, e(t + 1)$ and $r'^{t+1}$ using procedure NON-EQUIVALENCE;
end
else
begin
if there is an active number $k$ at $t + 1$
then
begin
$a'^{t+1} := k$ and $ForbZone'^{t+1} := FZ'(k, t)$;
Define $q'^{t+1}$ using procedure UNDECIDABILITY;
$e(t + 1) := \max(t + 1, e(t))$ and $r'^{t+1} := r'$
end
else
begin
$q'^{t+1} := \varphi, \; a'^{t+1} := a', \; ForbZone'^{t+1} := ForbZone', \; e(t + 1) := \max(t + 1, e(t))$ and $r'^{t+1} := r'$
end
end
\forall x \in [0, t] [v'^{t+1}(x) := \mu^{e(t+1)} \circ q'^{t+1}(x)], \forall x \in [t, \infty] [v'^{t+1}(x) := v'(z)]$ and $\eta'^{t+1} := \{ (x, y) : x, y \in [0, t] \& (q'^{t+1}(x), q'^{t+1}(y)) \in \text{Equiv}(\eta'^{t+1}) \}$
end of Stage $t + 1$

Where the procedures NON-EQUIVALENCE and UNDECIDABILITY are as described below.

Procedure NON-EQUIVALENCE
begin
(R1) /* Find $\varphi$ and $t'$ s.t. $\eta^t \subseteq \{ (x, y) : x, y < t \& (\varphi(x), \varphi(y)) \in \text{Equiv}(\eta^t) \}$
s.t. for all $x \in \omega$ if $x \in \text{Rng}(f^{t+1}_n) \setminus \text{ForbZone}^{t+1}$ then $v^t(x) \subseteq \mu^t(\psi(x))$ and otherwise $\psi(x) = q^t(x)$. */

begin
Let $x_1, \ldots, x_q \in \text{Rng}(f^{t+1}_n) \setminus \text{ForbZone}^{t+1}$ be s.t. $\text{Cl}'(x_1), \ldots, \text{Cl}'(x_q)$ are distinct and $\text{Rng}(f^{t+1}_n) \setminus \text{ForbZone}^{t+1} \subseteq \bigcup_{i \in [1, q]} \text{Cl}'(x_i)$; Find the least $t' \geq e(t)$ s.t. for every $i \in [1, q]$ at least one of the following two events will occur:

(NE1') For all $y \in \text{Cl}'(x_i)$, $v^t(y) \subseteq \mu^t'(\psi'(y))$ or (NE2') There is $z_i$ s.t. for all $y \in \text{Cl}'(x_i)$, $v^t(y) \subseteq \mu^t(z_i)$;

Define $\psi$ as follows. If for some $x \in \omega$ there is $i \in [1, q]$ s.t. $x \in \text{Cl}'(x_i)$ and (NE1') is not true for $i$ then $\psi(x) := z_i$, otherwise $\psi(x) := q^t(x)$;

end

(R2) /* Place $[0, a^{t+1})$ into the range of $q^{t+1}$ */

begin
Let $l$ be the number of elements in $[0, a^{t+1}) \setminus \text{Rng}(\psi)$; Define $q^{t+1}$ as follows. If for some $i \leq l$, $x$ is the $i$th element in $\omega \setminus ([0, t] \cup \text{ForbZone}^{t+1})$ then define $q^{t+1}(x)$ to be the $i$th element in $[0, a^{t+1}) \setminus \text{Rng}(\psi)$, otherwise $q^{t+1}(x) := \psi(x)$;

end

$r^{t+1}(m, n) := r^t(m, n) + 1$ and $\forall m', n' \in \omega [(m', n') \neq (m, n) \Rightarrow r^{t+1}(m', n') := r^t(m', n')]$;

end NON-EQUIVALENCE

Procedure UNDECIDABILITY

begin
Let $(x, z)$ be the $\prec$-least pair s.t. $x \notin \text{ForbZone}^{t+1}$, $\chi^{t+1}_k(x, z) = 0$ and $\forall y \in \text{Cl}'(x)$ $v^t(y) \subseteq v^t(z)$;

For all $y \in \text{Cl}'(x)$ $\psi(y) := q^t(z)$ and for all $y \notin \text{Cl}'(x)$ $\psi(y) := q^t(y)$;

(R3) /* Place $[0, a^{t+1})$ into the range of $q^{t+1}$ */

begin
Let $l$ be the number of elements in $[0, a^{t+1}) \setminus \text{Rng}(\psi)$; Define $q^{t+1}$ as follows. If for some $i \leq l$, $x$ is the $i$th element in $\omega \setminus ([0, t] \cup \text{ForbZone}^{t+1})$ then define $q^{t+1}(x)$ to be the $i$th element in $[0, a^{t+1}) \setminus \text{Rng}(\psi)$, otherwise $q^{t+1}(x) := \psi(x)$;

end

end UNDECIDABILITY

Lemma 5.1. If $v^t(x) = \mu^t(\psi(x))$ for $x \in [0, t)$ and $v^t(t) = \emptyset$ then the stage $t + 1$ terminates.

(Note that here we are using only (T2').)

Proof. We will show that (R1) terminates. It is easy to see that for all $y \in \text{Cl}'(x_i)$, $\mu(\psi'(y)) = \mu(q'(x_i))$. If $\mu(q'(x_i))$ is infinite then obviously there is some $i' \geq e(t)$ s.t. (NE1') is true for $t'$ and $i$. If $\mu(q'(x_i))$ is finite then by (T2') there is some $z$ s.t. $\mu(q'(x_i)) \subseteq \mu(z)$ and hence (NE2') will occur. □
Corollary 5.1. The requirements (1)-(4) from Lemma 4.2 are satisfied.

So by Lemma 4.2, $\nu$ is indeed a computable enumeration. We shall show the other required properties of $\nu$.

Corollary 5.2. If for some $t$, $y \in \text{Cl}'(x)$ then for all $t' \geq t$, $y \in \text{Cl}'(x)$ and $\nu(y) = \nu(x)$.

Proof. By observation of the algorithm it is easy to show that for every $t$, $x \in \omega$, $\varphi_{t+1}$ is either coincident with $\varphi_t$ over $\text{Cl}'(x)$ or is constant over $\text{Cl}'(x)$. Therefore for every $t$ s.t. $y \in \text{Cl}'(x)$, we have $y \in \text{Cl}'^{t+1}(x)$ and hence for all $t' \geq t$, $y \in \text{Cl}'(x)$. It also follows that $\forall t' \geq t \forall y \in \text{Cl}'(x)$ $[\varphi_t(y) = \varphi_t'(y)$ and $(\varphi_t'(y), \varphi_t'(x)) \in \eta_y]$ or $\exists t' \geq t \forall t'' \geq t' \forall y \in \text{Cl}'(x)$ $[\varphi_t''(x)]$. The last part of the corollary follows from this and (1) and (4) from Lemma 4.2. $\Box$

Corollary 5.3. $\{\eta^t_v\}_{t \in \omega}$ is a recursive approximating sequence of $\bigcup_{v \in \omega} \eta^t_v$ and $\bigcup_{v \in \omega} \eta^t_v \subseteq \eta_v$.

Corollary 5.4. The requirements (2), (3) and (5) from Lemma 4.1 are satisfied.

Corollary 5.5. For all $t$ if $(m, n)$ is active at $t+1$ then for all $x \in \text{Rng}(f_{n+1}'') - \text{ForbZone}^{t+1}$, $\varphi'(x) \subseteq \varphi''(x)$.

Lemma 5.2. The following hold.

1. For all $t$ and $x$, $\text{FZ}'(x, t) \cap \{0, t\}$ contains at most $2x$ equivalence classes at stage $t$.

2. For all $t$ there is $y \geq t$ s.t. for all $x \geq y$, $\varphi_t'(x) = x$.

Lemma 5.3. Each pair $(m, n)$ could be active only at finitely many stages.

Proof. Suppose $(m, n)$ is the $< - \text{minimal pair}$ which is active infinitely often. Then by the algorithm $r'(m, n) \rightarrow \infty$ with $i \rightarrow \infty$ and hence by (A2) $f_n$ is total. Therefore by (T1) there are distinct $y_1, \ldots, y_p \in \text{Dom}(f_n)$ with $p > \max(m, n)$ and s.t. $\gamma_m(y_1), \ldots, \gamma_m(y_p)$ are finite and distinct. From (A2), $r'(m, n) \rightarrow \infty$ and the hypothesis it easily follows that for all $i \in \{0, p\}$, $\gamma_m(y_i) = \nu(f_n(y_i))$ and therefore $\nu(f_n(y_i)), \ldots, \nu(f_n(y_p))$ are distinct.

Let $t$ be s.t. $(m, n)$ is active at $t+1$, $y_1, \ldots, y_p \in \text{Dom}(f_{n+1}')$ and for all $i \in \{0, p\}$, $\gamma_m(y_i) = \nu'(f_n(y_i)) = \nu(f_n(y_i))$. Then by Corollary 5.2, $\text{Cl}'(f_n(y_i)), \ldots, \text{Cl}'(f_n(y_p))$ are distinct. Therefore by Lemma 5.2 there is some $i \in \{1, p\}$ s.t. $f_n(y_i) \in \text{Rng}(f_{n+1}') - \text{ForbZone}^{t+1}$. By Corollary 5.5, $\varphi'(f_n(y_i)) \subseteq \varphi^{t+1}(f_n(y_i))$ and hence $\nu(f_n(y_i)) \subseteq \nu^{t+1}(f_n(y_i))$. Contradiction. $\Box$

Lemma 5.4. Each number $k$ could be active only at most once.
Lemma 5.5. There are infinitely many pairs or numbers which are active at least once.

Proof. Suppose the opposite. Then by Lemmas 5.3 and 5.4 there is \( t \) s.t. for all \( (m, n) \) and for all \( k \) neither \( (m, n) \) nor \( k \) is active at any \( t' \geq t \). Therefore \( \nu = \mu \circ \varphi' \). Let \( (m, n) \) be s.t. \( (m, n) \) is never active and \( \text{Rng}(f_n) = \omega \). So by (T1) and since by (2) of Lemma 5.2 for some \( t' \), \( [t', \infty) \subseteq \text{Rng}(\varphi') \) and \( \varphi' = \text{Id}_{(t', \infty)} \), we have that there are \( x_1, \ldots, x_p \in \text{Rng}(\varphi') \) with \( p > 2 \max(m, n) \) and s.t. \( \mu(x_1), \ldots, \mu(x_p) \) are distinct and hence \( \nu(x_1), \ldots, \nu(x_p) \) are distinct. Then by Corollary 5.2 for any \( t' \in \omega \), \( \text{Cl'}(x_1), \ldots, \text{Cl'}(x_p) \) are distinct. Since \( \text{Rng}(f_n) = \omega \), \( x_1, \ldots, x_p \in \text{Rng}(f_n) \). By Lemma 5.2 there is some \( t' > t \) s.t. \( \text{Rng}(f_n') - \text{ForbZone}' \neq \emptyset \) and hence (A3) holds at \( t' \) for \( (m, n) \). Without loss of generality we can assume that \( m, n < t' \) and therefore (A1) holds. Since \( (m, n) \) is never active (A2) also holds and therefore \( (m, n) \) is active at \( t' \). Contradiction. \( \square \)

Corollary 5.6. The requirement (1) from Lemma 5.1 holds.

So Lemmas 4.1 and 4.2 are applicable and therefore:

Corollary 5.7. \( \nu \) is a computable enumeration and there is a \( \varphi = \lim_{t \to \infty} \varphi' \) s.t. \( \varphi \) is surjective and \( \nu = \mu \circ \varphi \).

Lemma 5.8. \( \eta_\nu \) is r.e.

Proof. By Corollary 5.3, \( \bigcup_{t \in \omega} \eta_t' \subseteq \eta_\nu \) and \( \bigcup_{t \in \omega} \eta_t' \) is r.e. So it is enough to show that \( \eta_\nu \subseteq \bigcup_{t \in \omega} \eta_t' \). Since \( \nu = \mu \circ \varphi, \ \eta_\nu = \{ (x, y) : \mu \circ \varphi(x) = \mu \circ \varphi(y) \} \). So if \( \nu(x) = \nu(y) \) then \( \langle \varphi(x), \varphi(y) \rangle \in \eta_\nu \) and hence for some \( t \) and all \( t' \geq t \), \( \langle \varphi(x), \varphi(y) \rangle \in \eta_t \). Since by Corollary 5.7, \( \varphi = \lim_{t \to \infty} \varphi' \), for some \( t' \geq t \), \( \varphi(x) = \varphi'(x) \) and \( \varphi(x) = \varphi'(x) \) Therefore \( \langle x, y \rangle \in \eta_t \). \( \square \)

Corollary 5.8. \( \nu \) is a positive enumeration of \( S \) and hence \( \nu \in K \).

Lemma 5.7. If \( C \) is not empty then all \( m \in \omega \), \( \nu_m : \gamma \rightarrow v \).

Proof. Suppose \( \gamma_m = \nu \). Then for some recursive \( f \), we would have \( \gamma_m = \nu \circ f \). Since \( C \) is not empty then \( \gamma_m \in K \) and hence \( \nu \circ f \) is an enumeration of \( S \). By (T1) there are \( y_1, \ldots, y_p \) with \( p > 2 \max(m, n) \) and s.t. \( \nu(f_n(y_1)), \ldots, \nu(f_n(y_p)) \) are distinct. By Corollary 5.2 for any \( t \in \omega \), \( \text{Cl}(f_n(y_1)), \ldots, \text{Cl}(f_n(y_p)) \) are distinct. By Lemma 5.2 it is easy to see that there is some \( t \in \omega \) s.t. (A3') holds for \( (m, n) \) at all \( t' \geq t \). Without loss of generality we may consider that (A1) holds for \( (m, n) \) at all \( t' \geq t' \). By Lemma 5.3 we may also assume that no pair \( (m', n') < (m, n) \) is active at any \( t' \geq t \). Hence for any \( t' \geq t \), \( \text{r}(m, n) = \text{r}(m, n) \). Therefore since \( \gamma_m = \nu \circ f \), it is easy to see that there is \( t' \geq t \) s.t. (A2) holds for \( (m, n) \) at all \( t' \). So \( (m, n) \) is active at all \( t'' \geq t' \). Contradiction. \( \square \)
Lemma 5.8. $\eta_\nu$ is not recursive.

Proof. Suppose the opposite. Then some recursive $\chi_k$ is a characteristic function of $\eta_\nu$. This implies that $k$ is never active and that (AN2) holds for $k$ at all stages. By Corollary 5.8, $\nu$ is an enumeration of $S$. Thus by (T1) there are $y_1, \ldots, y_p$ with $p > 2k$ and s.t. $\nu(y_1), \ldots, \nu(y_p)$ are finite and distinct. By (T2') there are $z_1, \ldots, z_p$ s.t. for all $i \in [0, p]$, $\nu(y_i) \subset \nu(z_i)$. Then there is some $t > k$, $y_1, \ldots, y_p, z_1, \ldots, z_p$ s.t. for all $i \in [0, p]$, $\nu'(y_i) = \nu(y_i)$ and $\nu(y_i) \cap \nu'(z_i)$. Since by Lemma 5.2 there are no more than $2k$ equivalence classes in $FZ'(k, t) \cap [0, t)$ we have for some $i \in [0, p]$, $y \notin FZ'(k, t)$. It is easy to see that $\forall v \in Cl'(x) [\nu'(y) \subseteq \nu'(z)]$. Since $\chi_k$ is a characteristic function of $\eta_\nu$, $\chi_k(y_i, z_i) = 0$. We can assume without loss of generality that $\chi_k^{i+1}(y_i, z_i) = 0$. Therefore $k$ is active at $t + 1$. Contradiction. □

6. Negative undecided enumerations: a proof of Theorem 3

We will use the following notational conventions. If $R$ is a binary relation on $\omega$ then $\text{Sym}(R)$ designates the symmetric closure of $R$.

Assume that we are given a recursive approximating sequence $\{\mu'(x)\}_{t, x \in \omega}$ of a computable enumeration $\mu \in K_{\text{neg}}$, a recursive approximating sequence $\{\theta_\mu\}_{t \in \omega}$ of $\theta_\mu$ and a recursive approximating sequence $\{\gamma'(x)\}_{t, x \in \omega}$ of a computable class $C \subseteq K$. We would like to construct a computable enumeration $\nu \in K$ s.t. if $C$ is not empty then $\forall m = \nu$ for all $x \in \omega$.

6.1. Insuring that $\nu$ is negative and undecided

We shall guarantee that $\nu$ is a computable enumeration in the same way as in Section 4.

Further, we shall build $\nu$ in such a way that there is $\varphi: \omega \rightarrow \omega$ s.t. $\varphi$ is surjective and $\nu = \mu \circ \varphi$. This will guarantee that $\nu \in H(S)$. To guarantee $\nu \in K$ we will have to insure that $\nu$ is negative and not decidable. This is the same as insuring that $\theta_\nu$ is r.e. but not recursive.

We shall guarantee that $\theta_\nu$ is r.e. as follows. Assume that $e: \omega \rightarrow \omega$ is a monotonically increasing recursive function, $\nu = \mu \circ e \varphi$ and $\varphi = \lim_{t \rightarrow \omega} \varphi(t)$ where $\{\varphi(t)\}_{t \in \omega}$ is a sequence of recursive functions. Let us define a sequence $\{\theta_\nu\}_{t \in \omega}$ of finite sets by $\theta_\nu = \{x, y: x, y \in [0, t) \& (\varphi(t), \varphi'(t)) \in \text{Sym}(\theta_\mu(t))\}$. The presence of $e(t)$ in the definition of $\theta_\nu$ marks the difference between the proofs of Theorems 2 and 3. In distinction with Section 5, it is not enough to use $\theta_\mu$ at stage $t$ since the present algorithm at stage $t$ looks for an approximating sequence of $\theta_\mu$ satisfying certain additional properties (which we shall describe later). It is easy to see that this definition guarantees that $\{\theta_\nu\}_{t \in \omega}$ is a strong array and that therefore $\bigcup_{t \in \omega} \theta_\nu$ is r.e. (note that as in previous theorems $\nu'(x) = \mu'(t)(x)$ for all $x, t$). The next step is to insure that $\bigcup_{t \in \omega} \theta_\nu = \theta_\nu$. Since $\{\theta_\nu\}_{t \in \omega}$ is a recursive
approximating sequence of $\theta_\mu$ it is easy to show that $\theta_\nu \subseteq \bigcup_{t \in \omega} \theta_\nu^t$. All of the above together with $\forall \tau \in \omega \left[ \theta_\nu^t \subseteq \theta_\nu^{t+1} \right]$ would guarantee that $\bigcup_{t \in \omega} \theta_\nu^t \subseteq \theta_\nu$.

It is easy to see that in order to insure that for all $t \in \omega$ $\theta_\nu^t \subseteq \theta_\nu^{t+1}$, we have to define $\{ \varphi' \}_{t \in \omega}$ in such a way that for every $t$, $x \in \omega$,

$$\forall y \in [0, t] \left[ \langle x, y \rangle \in \theta_\nu^t \Rightarrow \langle \varphi^{t+1}(x), \varphi^{t+1}(y) \rangle \in \text{Sym}(\theta_\nu^{t+1}) \right].$$

We insure that $\theta_\nu$ is not recursive as follows. If $\theta_\nu$ were recursive then the characteristic function of $\theta_\nu$ would be a two-placed recursive function. So we will try to insure that for any $k$ the $k$th two-placed p.r. function $\chi_k$ is not a characteristic function of $\theta_\nu$. To do that it is enough to construct $\theta_\nu$ and $\chi_k$ in such a way that for any $t$, $x \in \omega$,

$$\langle x, y \rangle \in \theta_\nu \text{ and } \chi_k(x, y) = 0 \text{ or } (\langle x, y \rangle \notin \theta_\nu \text{ and } \chi_k(x, y) = 1).$$

In difference with the proof of Theorem 2, the present algorithm explicitly tries to achieve both clauses of this disjunction.

If we rewrite the two above clauses in terms of recursive approximating sequences of $\chi_k$ and of $\theta_\nu$, it is easy to notice that they are fundamentally different. The first clause is existential and has the following form:

For some stage $t \in \omega$, $\chi_k(x, y) = 0$ and $\langle x, y \rangle \in \theta_\nu^t$.

Once achieved, this condition cannot be injured since for all $t \in \omega$, $\theta_\nu^t \subseteq \theta_\nu^{t+1}$. The second clause is universal and has the following form:

For all stages $t \in \omega$, $\chi_k(x, y) = 1$ and $\langle x, y \rangle \notin \theta_\nu^t$.

It can be injured, since if for some stage $t$, $\chi_k(x, y) = 1$ and $\langle x, y \rangle \notin \theta_\nu^t$, then we cannot be sure that $\langle x, y \rangle$ will not be placed in $\theta_\nu$ at some later stage (unless $x = y$).

We are going to define two terminating procedures, $P_1$ and $P_2$ with the intent of satisfying the first and second clauses. At each stage dealing with $\chi_k$, we use both procedures.

Procedure $P_1$ works as follows. It attempts to put $\langle x, y \rangle$ into $\theta_\nu^{t+1}$ for every pair $(x, y)$ s.t. $\chi_k(x, y) = 0$, $x \neq y$, $\langle x, y \rangle \notin \theta_\nu^t$, and $(x$ or $y$ is not in ForbZone$^{t+1}$). The procedure will succeed in putting $\langle x, y \rangle$ into $\theta_\nu^{t+1}$ if $\mu \circ q'$ is finite at an element of a pair $(x, y)$, not contained in ForbZone$^{t+1}$ and it is equal to $\chi_k(x, y)$ at that point. If the algorithm succeeds at least on one pair $(x, y)$ of the described type, $\chi_k$ is not a characteristic function of $\theta_\nu$, by the first clause.

The procedure $P_2$ acts if $P_1$ has not succeeded and starts after $P_1$ has terminated. The procedure $P_2$ attempts to pair every $x < t$ s.t. $(x$ is not in ForbZone$^{t+1}$ and $P_2$ has not been successfully applied to $x$ at earlier stages while acting in respect to $k$), with some $y \neq x$ s.t. $q^{t+1}(x) = q^{t+1}(y)$. It will succeed in pairing for those $x$ at which $\mu \circ q'$ is finite and $\chi_k(x) = \mu \circ q'(x)$. After that, for all pairs $(x, y)$ where it succeeded, $P_2$ protects $(q'$) for further changes at both $x$ and $y$ by placing a marker $(x, y)$ on $k$. In the next paragraph we will explain in
Some effectively infinite classes of enumerations

229
details how the protection by markers works. For all those stages \( t' \) where the protection works we would have \( \varphi^{t'+1}(x) = \varphi^{t'+1}(y) \) and hence \( \langle x, y \rangle \notin \Theta_v' \). We intend the protection to be continued until a stage \( t' \) when we are again dealing with \( \chi_k \) (i.e., \( k \) is active), \( \chi_k'(x, y) = 0 \) and \( x, y < t' \) or otherwise indefinitely long. In the former case we shall apply \( P_i \) to \( (x, y) \) and in the latter case we have \( \langle x, y \rangle \notin \Theta_v' \). Also in the last case there are two possibilities. The first is \( \chi_k'(x, y) = 1 \) which means that \( \chi_k \) is not a characteristic function of \( \Theta_v \) by the second clause. The second possibility is that \( \chi_k \) is not defined on \( (x, y) \) which also means that \( \chi_k \) is not a characteristic function of \( \Theta_v \).

Now we shall describe the ‘protection’ by markers. Suppose at stage \( t + 1 \) there is an active pair \( (m, n) \) or an active number \( l \). If \( k < \max(m, n) \) or \( k < l \) then for all markers \( (x, y) \) which are on \( k \) at stage \( t \) we impose \( \varphi^{t'+1}(x) = \varphi'(x) \) and \( \varphi^{t'+1}(y) = \varphi'(y) \). If \( k \geq \max(m, n) \) or \( k > l \) then we allow \( \varphi^{t'+1}(x) \neq \varphi'(x) \) or \( \varphi^{t'+1}(y) \neq \varphi'(y) \), and in a case if one of the inequalities really occurred we would say that the marker \( (x, y) \) was injured and we would remove it from the number \( k \). If \( k = l \) and we are applying \( P_i \) then we treat all markers on \( k \) as if \( k > l \). Conversely, if we are applying \( P_2 \) then we treat them as if \( k < l \).

Now we will formulate conditions from which the above requirements on \( P_i \) follow. To simplify the description, we shall deal with only one pair \( (x, y) \) s.t. \( \chi_k'(x, y) = 0 \), \( x \neq y \) and \( \langle x, y \rangle \notin \Theta_v' \). As in proofs of Theorems 1 and 2 we may assume that for all \( t \) there is \( x, \) s.t. for all \( u \}, \varphi'(u) = u \). We shall look for some \( z \) s.t. \( \varphi'(z) = z \) with the intention to define \( \varphi^{t'+1}(x) = z \) and for all \( u \neq x, \varphi^{t'+1}(u) = \varphi'(u) \). In order to guarantee \( \varphi'(x) \subseteq \varphi^{t'+1}(x) \) we have to insure \( \varphi'(x) \subseteq \varphi^{t'+1}(z) \), and in order to guarantee \( \Theta_v' \subseteq \Theta_v^{t'+1} \) we have to insure that \( \forall u \in [0, t) \ \{ \langle x, u \rangle \in \Theta_v' \Rightarrow \langle z, \varphi'(u) \rangle \in \text{Sym}(\Theta_v^{t'+1}) \} \). Also, since we want \( \langle x, y \rangle \in \Theta_v^{t'+1} \), we have to insure that \( \langle z, \varphi'(y) \rangle \in \text{Sym}(\Theta_v^{t'+1}) \).

We shall demonstrate that the conditions of the preceding paragraph can be satisfied. Note that \( \varphi'(x) = \mu^{\varphi'(x)}(\varphi'(x)) \). If \( \mu(\varphi'(x)) \) is finite, by (T2) there are infinitely many such \( z \) that \( \mu(\varphi'(x)) \subseteq \mu(z) \) and hence for some \( t' \), \( \varphi'(x) \subseteq \mu^{t'}(x) \). Among those \( z \) as above there also infinitely many such \( z \)’s that \( \forall u \in [0, t') \ \{ \langle x, u \rangle \in \Theta_v' \Rightarrow \mu(\varphi'(u)) \neq \mu(z) \} \) and hence for some \( t'' \), \( \forall u \in [0, t') \ \{ \langle x, u \rangle \in \Theta_v' \Rightarrow \langle z, \varphi'(u) \rangle \in \Theta_v^{t''} \} \). Among these \( z \) we can take any \( z \geq x, \) s.t. \( \mu(\varphi'(y)) \neq \mu(z) \). Then for some \( t''' \), \( \langle z, \varphi'(y) \rangle \in \Theta_v^{t'''} \). So it is enough to define \( e(t + 1) = \max(e(t), t', t'', t''') \), \( \varphi^{t'+1}(x) = z \) and for any \( u \neq x, \varphi^{t'+1}(u) = \varphi'(u) \). The fact that there are such \( z \)’s gives us an effective way to find one of them. We can search for \( z \) with the required properties all components of the pairs in \( \Theta_v^{(t)} \), \( \Theta_v^{(t'+1)} \), . . . until we find one. Since every \( \Theta_v' \) is finite, there are only finite numbers of computations at each step and therefore the process will terminate. Since we do not know in advance whether \( \mu(\varphi'(x)) \) is finite and \( \varphi'(x) = \mu \circ \varphi'(x) \), it is possible that a \( z \) as above may not exist and therefore the procedure as described so far may not terminate. To guarantee termination we stop the procedure whenever any of the following two events will occur. One event is the discovery of a desired \( z \) inside the sequence \( \Theta_v^{(t'+1)} \), described above. The other event is the discovery
that $\nu'(x) \subseteq \mu(\nu'(x))$ by seeing that $\nu'(x) \subseteq \mu^{(t+1)}(\nu'(x))$ after we have failed to find suitable $z$ in $\theta_{\nu}^{(t+1)}$. Since the family $S$ satisfies (T1) and (T2) one of the two events has to occur and the procedure will terminate.

Now we will formulate conditions from which the above requirements on $P_2$ follow. For simplicity we shall deal with only one $x \leq t$. We shall look for some $y \neq x$ s.t. $\nu'(y) = y$ with the intention to define $\nu^{t+1}(x) = y$ and for all $u \neq x$, $\nu^{t+1}(u) = \nu'(u)$. It is easy to see that this gives us $\nu^{t+1}(x) = \nu'(y)$. After that we shall place the marker $\{x, y\}$ upon $k$. In order to guarantee $\nu'(x) \subseteq \nu^{t+1}(x)$ we have to insure $\nu'(x) \subseteq \mu^{(t+1)}(y)$, and in order to guarantee $\theta_{\nu} \subseteq \theta_{\nu}^{t+1}$ we have to insure that $\forall u \in [0, t)$ $[\langle x, u \rangle \in \theta_{\nu} \Rightarrow \langle y, \nu'(u) \rangle \in \text{Sym}(\theta_{\nu}^{(t+1)})]$. The proof that these conditions can be satisfied is analogous to the one for $P_1$.

6.2. The algorithm

For all stages $t$ the function $d' : \omega \to \omega$ serves to reject the functions $\chi_k$ which are not total or s.t. $\chi_k \mid (d'(k) \times d'(k))$ is not the characteristic function of $\theta_{\nu}$ restricted to the square $d'(k) \times d'(k)$. It plays a similar role to that of the function $r'$ in respect to functions $f_n$.

**Definition.** 1. Mark($l$, $t$) = \{ $x \in [0, t)$: $\exists y \in [0, e(t))$ \}$l$ is marked by $\{x, y\}$ at stage $t$\}.

2. CumMark($l$, $t$) = $\bigcup_{k<l} \text{Mark}(k, t)$.

3. A pair $(m, n)$ requires attention at $t + 1$ if (A1), (A2) and
   (A3') $\text{Rng}(f_{\nu}^{t+1}) - (\text{FZ}(\text{max}(m, n), t) \cup \text{CumMark}(\text{max}(m, n), t)) \neq \emptyset$.

4. A number $k$ requires attention at $t + 1$ if (AN1) and
   (AN2') $\text{Dom}(\chi_k^{t+1}) \ni \langle x, y \rangle \in \{0, 1\}$ and $(\forall x, y \in [0, t)$ $\chi_k^{t+1}(x, y) = 0 \Rightarrow \langle x, y \rangle \notin \theta_{\nu}^{t+1})$ and $(\forall x, y \in [0, d'(k)) \chi_k^{t+1}(x, y)$ is defined and $\chi_k^{t+1}(x, y) = 1 \Leftrightarrow \langle x, y \rangle \in \theta_{\nu}^{t+1}$; and
   (AN3') The following disjunction is true.

   (AN3.1) $\{\langle x, y \rangle \mid x, y \in [0, t), x \neq y, (x \notin \text{FZ}(k, t) \cup \text{CumMark}(k, t)) \text{ or } y \notin \text{FZ}(k, t) \cup \text{CumMark}(k, t), \chi_k^{t}(x, y) = 0 \text{ and } \langle x, y \rangle \notin \theta_{\nu}^{t+1}\} \neq \emptyset$ or
   (AN3.2) $\langle x, y \rangle \notin \langle x, y \rangle \in \theta_{\nu}^{t+1}$.

5. A pair $(m, n)$ is active at $t + 1$ if $t + 1$ is odd and $(m, n)$ is the $\leq$-least of all pairs which requires attention at $t + 1$, where $(m, n) \leq (m', n')$ if $\langle m, n \rangle \leq \langle m', n' \rangle$.

6. A number $k$ is active at $t + 1$ if $t + 1$ is even and $k$ is the least of all numbers which requires attention at $t + 1$.

**Construction**

Stage 0. $\nu^0 := 1d_\omega$, $\forall x \in \omega, \nu^0(x) = \emptyset$, $\theta_{\nu}^{0} = \emptyset$, $a' := 0$, $\text{ForbZone}' := \emptyset$, $e(t) := 0$, $\forall m, n \in \omega, \nu^0(m, n) := 0$, $\forall k \in \omega, d'(k) := 0$ and no numbers are marked.

Stage $t + 1$. 
begin
  if there is an active pair \((m, n)\) at \(t + 1\) then
  begin
    \(a^{t+1} := \max(m, n)\) and \(\text{ForbZone}^{t+1} := \text{FZ}(\max(m, n), t) \cup \text{CumMark}(\max(m, n), t)\); Define \(\varphi^{t+1}, e(t + 1), r^{t+1}\) and the distribution of markers using procedure NON-EQUIVALENCE';
    \(d^{t+1} := d^t;\)
  end
else
  if there is an active number \(k\) at \(t + 1\) then
  begin
    \(a^{t+1} := k\) and \(\text{ForbZone}^{t+1} := \text{FZ}(k, t) \cup \text{CumMark}(k, t)\); Define \(\varphi^{t+1}, e(t + 1), d^{t+1}\) and the distribution of markers using procedure UNDECIDABILITY';
    \(r^{t+1} := r^t;\)
  end
else
  begin
    \(q^t := q, a^{t+1} := a^t, \text{ForbZone}^{t+1} := \text{ForbZone}^t, e(t + 1) := \max(t + 1, e(t)), r^{t+1} := r^t, d^{t+1} := d^t\) and leave the distribution of markers unchanged;
  end;
end of Stage \(t + 1\)

Where the procedures NON-EQUIVALENCE' and UNDECIDABILITY' are as described below.

Procedure NON-EQUIVALENCE'
begin
  (R1) /* Find \(\psi\) and \(t'\) s.t. \(\theta'_t \subseteq \{\langle x, y \rangle: x, y < t \& \langle \psi(x), \psi(y) \rangle \in \text{Sym}(\theta'_t)\}\) and s.t. for all \(x \in \omega\) if \(x \in \text{Rng}(f^{t+1}_n) - \text{ForbZone}^{t+1}\) then \(\psi(x) \subseteq \mu'(\psi(x))\) and otherwise \(\psi(x) = \varphi'(x)\). */
begin
  Let \(\{x_1, \ldots, x_q\} = \text{Rng}(f^{t+1}_n) - \text{ForbZone}^{t+1}\), where \(x_1, \ldots, x_q\) are distinct, Find the least \(t' \geq e(t)\) s.t. there are \(z_1, \ldots, z_q \in [0, t')\) s.t. for every \(i \in [1, q]\) at least one of the following holds:
  (NE1') \(x_i = z_i\) and \(\varphi'(x_i) \subseteq \mu'(\varphi'(z_i))\) or
  (NE2') \(x_i \neq z_i, \varphi'(x_i) \subseteq \mu'(z_i), z_i = \varphi'(z_i)\) and for all \(y \in [0, t)\) if \(\langle x_i, y \rangle \in \theta'_t\) then \(y \neq z_i\) and for all \(y \in [0, t)\) if \(\langle z_i, y \rangle \in \text{Sym}(\theta'_t)\) and otherwise if for some \(j \in [1, q]\), \(y - z_j\) then \(\langle z_j, \varphi'(z_j) \rangle \in \text{Sym}(\theta'_t)\);
Define $q$ as follows. If for some $x \in \omega$ there is $i \in [1, q]$ s.t. $x = x_i$ and $x_i \neq z_i$ then $\psi(x) := z_i$ and otherwise $\psi(x) := \varphi'(x)$;

end;

(R2) /* Remove all the injured markers. Note that only markers placed on numbers greater or equal then max$(m, n)$ could be injured. */

begin

For all $i \in [1, q]$ s.t. $x_i \neq z_i$, all $l \in [\max(m, n), t)$ and all $y \in [0, e(t))$ if $l$ is marked by $\{x_i, y\}$ then remove the marker $\{x_i, y\}$ from $l$;

end

(R3) /* Place $[0, a'^{t+1})$ into the range of $q'^{t+1}$. */

begin

Let $l$ be the number of elements in $[0, a'^{t+1}) - \text{Rng}(\psi)$; Define $q'^{t+1}_l$ as follows. If for some $i \leq l$, $x$ is the $i$th element in $\omega - ([0, t] \cup \text{ForbZone}'^{t+1})$ then define $q'^{t+1}(x)$ to be the $i$th element in $[0, a'^{t+1}) - \text{Rng}(\psi)$, otherwise $q'^{t+1}(x) := \psi(x)$;

end

e(t + 1) := \max(t + 1, t'), r'^{t+1}(m, n) := r'(m, n) + 1 and

$\forall m', n' \in \omega \ [m', n') \neq (m, n) \Rightarrow r'^{t+1}(m, n) := r'(m, n)$;

end NON-EQUIVALENCE'

Procedure UNDECIDABILITY'

begin

if (A3.1) holds for $k$ /* then execute procedure $P_1$ (see Section 6.1). */ then

begin

(R4) /* Try to find $\psi$ and $t'^{e(t)}$ s.t. $\theta'_u \subseteq \langle x, y \rangle: x, y < t \& \langle \psi(x), \psi(y) \rangle \in \text{Sym}(\theta'_u) \rangle$ and s.t. for some $x, y \in [0, t]$, $\chi'^{t+1}(x, y) = 0$ and $\langle \psi(x), \psi(y) \rangle \in \text{Sym}(\theta'_u)$ (and hence $(x, y) \in \theta'^{t+1}_u$). The latter will definitely happen if for some $x_i$ as defined below $\mu(q'(x_i))$ is finite and $n(x_i) = \#(x_i)$. If this happens then $\chi_k$ cannot be a characteristic function of $\theta_v$ and by (AN2') $k$ will never be active afterwards. */

begin

Let $\text{Pairs} = \{\{x, y\}: x, y \in [0, t], x \neq y, (x \notin \text{ForbZone}'^{t+1} \text{ or } y \notin \text{ForbZone}'^{t+1}), \} \{(x, y) = 0\}$, $\{x_1, \ldots, x_q\} = \{x: x \in [0, t] - \text{ForbZone}'^{t+1}, \exists y \ [\{x, y\} \in \text{Pairs}]\}$, where $x_1, \ldots, x_q$ are distinct and let $y_1, \ldots, y_q$ be s.t. for all $i \in [1, q]$, $y_i = \min\{y: \{x_i, y\} \in \text{Pairs}\}$; Find the least $t'^{e(t)}$ s.t. there are $z_1, \ldots, z_q$ s.t. for every $i \in [1, q]$ at least one of the following holds:

(U1) $x_i = z_i$ and $n'(x_i) \subseteq \mu'(q'(x_i))$ or

(U2) $x_i \neq z_i$, $n'(x_i) \subseteq \mu'(z_i)$, $z_i = \varphi'(z_i)$, $\langle z_i, \psi(y_i) \rangle \in \text{Sym}(\theta'_u)$ and for all $y \in [0, t]$ s.t. $\langle x_i, y \rangle \in \theta'_u$ we have if $y \notin \{x_1, \ldots, x_q\}$ then $\langle z_i, \psi(y) \rangle \in \text{Sym}(\theta'_u)$ and otherwise if for some $j \in [1, q]$,
Some effectively infinite classes of enumerations

233

\[ y = z_i, \text{ then } \langle z_i, \psi(z_i) \rangle \in \text{Sym}(\theta'_n) \]

Define \( \psi \) as follows. If for some \( x \in \omega \) there is \( i \in [1, q] \) s.t. \( x = x_i \), then \( \psi(x) := z_i \), and otherwise \( \psi(x) := \varphi'(x) \).

end;

(R5) /* Remove all the injured markers. Note that only markers placed on
numbers greater than or equal to \( k \) could be injured. However, if a
marker placed upon \( k \) was injured at the previous step then \( \chi_k \) cannot be
a characteristic function of \( \theta_n \) and \( k \) will never be active afterwards.
Therefore in this case we do not need to maintain the markers placed
upon \( k \). */

begin

For all \( i \in [1, q] \) s.t. \( x_i \neq z_i \), all \( l \in (k, l) \) and all \( y \in [0, e(t)] \) if \( l \) is
marked by \( \{x_i, y\} \) then remove the marker \( \{x_i, y\} \) from \( l \);
If a marker placed upon \( k \) was injured at the previous step then
remove all the markers from \( k \);

end

else

begin

\( \psi := q' \) and \( t' := e(t) \);

end;

if (A3.2) holds and no marker placed upon \( k \) was injured at the previous step
/* then execute procedure \( P_2 \) (see Section 6.1). */ then

begin

(R6) /* Try to find \( q'^{+1} \) and \( t'^{+1} \) s.t. \( \{\langle x, y \rangle : x, y \in [0, l] \} \cap \langle \psi(x), \psi(y) \rangle \in \text{Sym}(\theta'_n) \}
\subseteq \{\langle x, y \rangle : x, y \in [0, l] \} \cap \langle \varphi'^{+1}(x), \varphi'^{+1}(y) \rangle \in \text{Sym}(\theta'_n) \}
\) and
s.t. for as many as possible pairs \( x, y \) s.t. \( x \in [0, l] \) and \( x \in (k, l) \) \( \cup \) \( \text{Mark}(k, t) \), \( y \in [0, t') \) and \( x \neq y \) we would have \( \varphi'^{+1}(x) = \varphi'^{+1}(y) \).
In these cases we shall protect the equality \( \varphi'^{+1}(x) = \varphi'^{+1}(y) \) by marking \( k 
with \( \{x, y\} \). */

begin

Let \( \{x_1, \ldots, x_q\} = [0, l] \) \( \setminus (\text{ForbZone}^{+1} \cup \text{Mark}(k, t)) \), where \( x_1, \ldots, x_q \) are distinct; Find the least \( t'' \geq t' \) s.t. there are \( z_1, \ldots, z_q \) s.t.
for every \( i \in [1, q] \) at least one of the following holds:

(U3) \( x_i = z_i \) and \( \varphi'(x_i) \subseteq \mu''(\psi(x_i)) \) or

(U4) \( x_i \neq z_i \), \( \varphi'(x_i) \subseteq \mu''(z_i) \), \( z_i = \psi(z_i) \) and for all \( y \in [0, t] \) s.t.
\( \langle \varphi(x_i), \psi(y) \rangle \in \text{Sym}(\theta'_n) \) we have if \( y \notin \{x_1, \ldots, x_q\} \) then
\( \langle z_i, \psi(y) \rangle \in \text{Sym}(\theta'_n) \) and otherwise if for some \( j \in [1, q] \),
y = z_j, then \( \langle z_i, \psi(z_j) \rangle \in \text{Sym}(\theta'_n) \);

Define \( \varphi'^{+1} \) as follows. If for some \( x \in \omega \) there is \( i \in [1, q] \) s.t. \( x = x_i \)
then \( \varphi'^{+1}(x) := z_i \), and otherwise \( \varphi'^{+1}(x) := \varphi(x) \);

end;
(R7) /* Remove all the injured markers. Note that only markers placed on numbers greater than \( k \) could be injured. */
\[
\text{begin}
\begin{align*}
\text{For all } i \in [1,q] \text{ s.t. } x_i \neq z_i, \text{ all } l \in (k,t) \text{ and all } y \in [0,e(t)) \text{ if } l \text{ is marked by } \{x_i,y\} \text{ then remove the marker } \{x_i,y\} \text{ from } l; \\
\end{align*}
\text{end;}
\]

(R8) /* Place the markers upon \( k \). */
\[
\text{begin}
\begin{align*}
\text{For all } i \in [1,q] \text{ s.t. } x_i \neq z_i \text{ place the marker } \{x_i,z_i\} \text{ upon } k.
\end{align*}
\text{end}
\]

else
\[
\text{begin}
\begin{align*}
q^{t+1} := q^t \text{ and } t^\prime := t^\prime;
\end{align*}
\text{end}
\]

(R9) /* Place \([0,a^{t+1})\) into the range of \( q^{t+1} \). */
\[
\text{begin}
\begin{align*}
\text{Let } l \text{ be the number of elements in } [0,a^{t+1}) - \text{Rng}(\psi); \text{ Define } q^{t+1} \text{ as follows. If for some } i < l, x \text{ is the } i\text{th element in } \omega \setminus ([0,l] \cup \text{ForbZonc}^{t+1} \cup \text{Mark}(k+1,t)) \text{ then define } q^{t+1}(x) \text{ to be the } i\text{th element in } [0,a^{t+1}) - \text{Rng}(\psi), \text{ otherwise } q^{t+1}(x) := \psi(x);
\end{align*}
\text{end;}
\begin{align*}
e(t+1) := \max(t+1, t^\prime), \ d^{t+1}(k) := d'(k)+1 \text{ and } \forall k' \neq k \ [d^{t+1}(k') := d'(k')];
\end{align*}
\text{end}
\]

UNDECIDABILITY'.

To show the correctness of this algorithm it is possible to use a similar sequence of lemmas and corollaries as in Sections 4 or 5 omitting those that refer to the equivalence classes defined there.

7. Friedberg enumerations and Pour-El and Howard's height function: a proof of Theorem 4

The conditions (H) and (C) are exactly the premises of Theorem 1 from [9]. Hence \( S \) has a Friedberg enumeration. Using this fact we may apply the method of proof from our Theorem 1 to show effective infinity of \( K \). We indicate below the needed changes in the algorithm from our Theorem 1 and in its correctness proof. We obtain the new algorithm from the old one as follows. Suppose \( h \) is a height function for \( S \) (see the definition at the end of Section 2.). Then the only required change is to replace the two predicates (NE1) and (NE2) in the old algorithm by one predicate (NE2) \& \( h(v'(x)) < h(\mu'(z)) \). We shall refer to the predicate as (NE).
The proof of correctness of the old algorithm uses assumptions (T1) or (T2) on the family S in only three lemmas. So, we have to modify the proofs of only these lemmas to obtain a correctness proof of the new algorithm. We have to show that the conclusions of the lemmas still hold if we use the existence of the height function for S instead of (T1) or (T2) of Theorem 1. The lemmas under consideration are 4.3, 4.4 and 4.6. Lemma 4.6 does not use (T2) and it uses only part of (T1), namely the fact that S is infinite. Since S has an enumeration, then \( S \neq \emptyset \). Hence the existence of a height function implies by its properties 2 and 3 that S is infinite. Therefore Lemma 4.6 does not require any changes. Lemma 4.3 is easy to modify using condition (3) from the definition of the height function and Lemma 4.4 can be modified based on the contradiction which employs the ascending chain condition of the height function and \( h(v''(x)) < h(\mu''(z)) \) from (NE).

Acknowledgement

We all would like to express our gratitude to Professor Anil Nerode for his encouragement and support.

References