On the Finiteness of the Recursive Chromatic Number

William I Gasarch * Andrew C.Y. Lee†

Abstract

A *recursive graph* is a graph whose vertex and edges sets are recursive. A *highly recursive graph* is a recursive graph that also has the following property: one can recursively determine the neighbors of a vertex. Both of these have been studied in the literature. We consider an intermediary notion: Let A be a set. An A-*recursive graph* is a recursive graph that also has the following property: one can recursively-in-A determine the neighbors of a vertex. We show that, if A is r.e. and not recursive, then there exists A-recursive graphs that are 2-colorable but not recursively k-colorable for any k. This is false for highly-recursive graphs but true for recursive graphs. Hence A-recursive graphs are closer in spirit to recursive graphs then to highly recursive graphs.

Keywords: Recursive graph, Highly recursive graph, recursive coloring.

AMSI codes: 03D25

1 Introduction

This paper grows out of an attempt to understand the finiteness of the recursive chromatic number of different classes of recursive graphs. A *recursive graph* $G = (V, E)$ is an infinite graph such that the vertex set V and the edge set E are recursive and every node has finite degree [Bea76]. The *recursive chromatic number* $\chi^r(G)$ is defined as the least number k such that there exists a recursive function f such that f is a proper coloring of the vertices of G with range $\{1, \ldots, k\}$ [Bea76]. In [Bea76], Bean shows that there exists a planar, connected recursive graph which is 3-colorable but not recursively k-colorable for any natural number k. By an easy modification [Sch80], one can construct a planar recursive graph which is 2-colorable but not recursively k-colorable for any natural number k. However, the graph is not connected in that case.

* Department of Computer Science and Institute of Advanced Computer Studies, University of Maryland, College Park MD 20742

† Department of Mathematics, University of Maryland, College Park MD 20742

The neighbor function $nbd(G)$ of a recursive graph G is the function which will, given a vertex v of G, return the set of neighbors of v. Note that if G is a recursive graph then $nbd(G) \leq_{\mathrm{T}} K$, where K is a complete r.e. set. A recursive graph G is *highly recursive* if $nbd(G)$ is recursive [Bea76]. It is known that [Sch80, CP83] if G is highly recursive then $\chi^r(G) \leq 2\chi(G) - 1$ and the bound is tight [Sch80]. Equivalently, we may restate the above results as

$$nbd(G) \leq_{\mathrm{T}} \emptyset \Rightarrow \chi^r(G) \leq 2\chi(G) - 1$$
$$nbd(G) \leq_{\mathrm{T}} K \Rightarrow \text{(no information)}.$$

A natural question is to see, for various sets A with $\emptyset <_{\mathrm{T}} A <_{\mathrm{T}} K$, whether or not $nbd(G) \leq_{\mathrm{T}} A$ implies any finite bound on $\chi^r(G)$. In this paper we prove that Bean's result can be generalized to the class of recursive graphs whose neighbor function is recursive in A, where A is any non-recursive r.e. set. Such graphs are called *A-recursive*. Formally, we prove that if A is a non-recursive r.e. set, then there exists an A-recursive graph G such that G is 2-colorable but not recursively k-colorable for any natural number k.

2 Notations and Basic Definitions

Let $\{e\}$ denote the e^{th} Turing Machine via a standard enumeration and $\langle -, - \rangle$ denote a recursive bijection from $\mathsf{N} \times \mathsf{N}$ to N. $\{e\}_s$ denotes the machine that runs $\{e\}$ for s steps. We write $\{e\}_s(z)\downarrow$ if on input z the machine $\{e\}$ converges to a value within s steps and $\{e\}(z)\downarrow$ if $(\exists s)(\{e\}_s(z)\downarrow)$. For any $X \subseteq \mathsf{N}, \{e\}(X) = \{y \in X : \{e\}(y)\downarrow\}$. The symbol $\exists^\infty$ is intended to be read as "there exist infinitely many".

We begin with a brief review on the notions of recursive graphs.

Definition 1 A graph $G = (V, E)$ is said to be *recursive* if the following hold:

1. $V \subseteq \mathsf{N}$ and $E \subseteq \mathsf{N} \times \mathsf{N}$ are recursive.
2. Every node of G has a finite number of neighbors.

In addition, let $nbd(G)$ denote the function which, given a node v of G, produces all the neighbors of v. If $nbd(G)$ is recursive, then the corresponding graph is said to be *highly recursive*.

We refine the above two notions by introducing A-recursive graphs.

Definition 2 Let A be any set. A graph $G = (V, E)$ is *A-recursive* if G is recursive and $nbd(G) \leq_{\mathrm{T}} A$.

Note that every recursive graph is K-recursive and every highly recursive graph is $\emptyset$-recursive. Our interest will be graphs that are A-recursive with $\emptyset <_{\mathrm{T}} A <_{\mathrm{T}} K$. For such graphs our main interest is to see whether their recursive chromatic number can be uniformly bounded. The main theorem answers this question for nonrecursive r.e. sets A. Its proof combines a variant of Bean's original construction with a permitting argument. The former enables us to show that the graph is 2-colorable but not recursively k-colorable for any k. The latter allows us to show that the neighbor function is recursive in the r.e. set A. For a discussion of the permitting method, see [Soa87].

3 Main Theorem

Theorem 3

Let A be a non-recursive r.e. set. There exists an A-recursive graph G such that G is 2-colorable but not recursively k-colorable for any natural number k.

Proof: The major part of this proof is the construction of a recursive graph G which satisfies the required conditions. Recall that $\langle -, - \rangle$ denotes a recursive bijection from $\mathsf{N} \times \mathsf{N}$ to N. We first recursively partition the set of natural numbers as $\mathsf{N} = \bigcup_{e,i \in \mathsf{N}} X_{\langle e,i \rangle} \cup X_{-1}$, where all X's are infinite. For each $\langle e, i \rangle \geq 0$, let $R_{\langle e,i \rangle}$ be the requirement:

$$R_{\langle e,i \rangle}: \quad \{e\} \text{ total} \Rightarrow \{e\} \text{ is not an } i\text{-coloring of } G.$$

Each requirement $R_{\langle e,i \rangle}$ is associated with a family of finite graphs $\{L_{\langle e,i \rangle}(j)\}_{j \geq 0}$, where each $L_{\langle e,i \rangle}(j)$ is a graph with 2^i isolated vertices. The vertices of the graphs $L_{\langle e,i \rangle}(j)$, $j \geq 0$, are obtained from a recursive partition of $X_{\langle e,i \rangle}$ into an infinite number of sets of size 2^i. All other vertices that are introduced during the construction come from X_{-1} in increasing order.

Our construction will proceed in stages. Let $A = \cup_{s=0}^{\infty} A_s$ be a fixed enumeration of A. We denote by $L^s_{\langle e,i \rangle}(j)$ the j^{th} finite graph associated with $R_{\langle e,i \rangle}$ at stage s, with $L^0_{\langle e,i \rangle}(j) = L_{\langle e,i \rangle}(j)$. We shall see that during the construction $L^s_{\langle e,i \rangle}(j) \neq L^{s+1}_{\langle e,i \rangle}(j)$ at finitely many stages s. Hence the subgraph $T_{\langle e,i \rangle}(j) = \cup_{s=0}^{\infty} L^s_{\langle e,i \rangle}(j)$ will be finite. The resulting graph after stage s is denoted by G_s. Then our final graph will be $G = \cup_{s \geq 0} G_s$ or equivalently, $G = \cup \{T_{\langle e,i \rangle}(j) : e, i, j \geq 0\}$. Note that G is not connected in our case.

G can be viewed as the union of an infinite array of finite graphs $T_{\langle e,i \rangle}(j)$, $e, i, j \geq 0$. For each $e, i, j \geq 0$, $T_{\langle e,i \rangle}(j)$ can be thought of as a finite graph with (possibly) several layers of vertices. The sequence of subgraphs $\{T_{\langle e,i \rangle}(j)\}_{j \geq 0}$ is intended to witness the fact that $\{e\}$ is not an i-coloring of G, while there is still some way to 2-color G.

3.1 Construction of G

Recall that A is a r.e. set. We assume that $\{a_0, a_1, \ldots\}$ is a recursive enumeration of A.

Let $s, e, i, j \in \mathsf{N}$. We will now define two different notions: what it means for the subgraph $L^s_{\langle e,i\rangle}(j)$ to *need attention* and what it means for $L^s_{\langle e,i\rangle}(j)$ to *deserve attention*. Just because the subgraph $L^s_{\langle e,i\rangle}(j)$ needs attention does not mean that it will get it. It has to also deserve attention. $L^s_{\langle e,i\rangle}(j)$ will deserve attention if A permits it to.

Definition 4 We say the finite subgraph $L^s_{\langle e,i\rangle}(j)$ *needs attention at stage* $s+1$ when $\{e\}_s(z)\downarrow$ for all vertices z in the subgraph. In addition, if

$$a_s \leq \max\{z : z \in L^s_{\langle e,i\rangle}(j)\},$$

then we say $L^s_{\langle e,i\rangle}(j)$ *deserves attention with A's permission at stage* $s+1$.

So, for any s, e, i, j, $L^s_{\langle e,i\rangle}(j)$ will belong to one of the following categories:

(a) $L^s_{\langle e,i\rangle}(j)$ does not need attention at stage $s+1$.

(b) $L^s_{\langle e,i\rangle}(j)$ needs attention at stage $s+1$ but $L^s_{\langle e,i\rangle}(j)$ does not deserve any attention at stage $s+1$. It is clear that $L^s_{\langle e,i\rangle}(j)$ may deserve attention at some later stages. This depends on our enumeration of the r.e. set A.

(c) $L^s_{\langle e,i\rangle}(j)$ deserves attention during stage $s+1$.

Construction

Stage 0**:** Set

1) $L^0_{\langle e,i\rangle}(j) = L_{\langle e,i\rangle}(j) \quad \forall e, i, j \geq 0$;

2) $G^0 = \bigcup\{L^0_{\langle e,i\rangle}(j) : e, i, j \geq 0\}$.

All requirements are declared unsatisfied.

Stage $s+1$**:** Compute A_{s+1}. Examine all the requirements $R_{\langle e,i\rangle}$ with $\langle e, i\rangle \leq s$ which are not yet satisfied and attempt to satisfy each of these $R_{\langle e,i\rangle}$ by working with the following $s+1$ finite graphs:

$$L^s_{\langle e,i\rangle}(0),\ L^s_{\langle e,i\rangle}(1), \ldots,\ L^s_{\langle e,i\rangle}(s).$$

For each such $\langle e, i\rangle$ we do the following:

1. For each j such that $L^s_{\langle e,i\rangle}(j)$ does not deserve attention, define $L^{s+1}_{\langle e,i\rangle}(j) = L^s_{\langle e,i\rangle}(j)$.

2. For each j such that $L^s_{\langle e,i\rangle}(j)$ deserves attention, apply the procedure stated in part 2 of Lemma 10 to $L^s_{\langle e,i\rangle}(j)$ once. Denote this expansion by $L^{s+1}_{\langle e,i\rangle}(j)$.

3. Declare $R_{\langle e,i\rangle}$ satisfied when it in fact becomes satisfied. Note that it is possible to effectively determine when $R_{\langle e,i\rangle}$ becomes satisfied.

Set $G^{s+1} = \bigcup\{L^{s+1}_{\langle e,i\rangle}(j) : e,i,j \geq 0\}$. Go to stage $s+2$.

End of Construction

The remaining part of the proof will be divided into three lemmas. First we show that G is 2-colorable. Second, we give an algorithm for $nbd(G)$ which is recursive in A. Finally we demonstrate that all requirements are satisfied.

Lemma 5 *G is planar and 2-colorable.*

Proof: It suffices to note that $G = \cup\{T_{\langle e,i\rangle}(j) : e,i,j \geq 0\}$. Each $T_{\langle e,i\rangle}(j)$ is obtained from $L_{\langle e,i\rangle}(j)$ through a finite number of expansions and via part 2 of Lemma 10. As all L_k's ($k \leq r$) are planar and 2-colorable, so are all of the $T_{\langle e,i\rangle}(j)$'s and therefore G is planar and 2-colorable. ∎

Lemma 6 $nbd(G) \leq_{\mathrm{T}} A$

Proof: We state the following algorithm for $nbd(G)$.

Algorithm for $nbd(G)$

(This algorithm returns all the neighbors of vertex x in graph G via queries to A.)

1. Input x.

2. By running the construction, find e,i,j,s such that $x \in L^{s}_{\langle e,i\rangle}(j) - L^{s-1}_{\langle e,i\rangle}(j)$.

   ($L^{-1}_{\langle e,i\rangle}(j)$ is the empty graph, by convention. This step is to find e,i,j so that $x \in T_{\langle e,i\rangle}(j)$.)

3. We focus on the subgraph $L^{s}_{\langle e,i\rangle}(j)$ and search for a stage s' such that if y is a neighbor of x in G then y is a neighbor of x in $G_{s'}$.

   Set $t = s$.

   Loop:

   a) Compute $m(t) = \max\{z\colon z \text{ is a vertex of } L^{t}_{\langle e,i\rangle}(j)\}$.

   b) (via query to A) Find $n(t) =$ the least stage t' such that

   $$A_{t'} \cap \{0,1,...,m(t)\} = A \cap \{0,1,...,m(t)\}.$$

   c) Run the construction up to stage $n(t)$.

d) If $L^t_{\langle e,i\rangle}(j) = L^{n(t)}_{\langle e,i\rangle}(j)$, then set $s' = n(t)$ and exit the loop.

e) If $L^t_{\langle e,i\rangle}(j) \neq L^{n(t)}_{\langle e,i\rangle}(j)$, then set $t = n(t)$ and go to step *a*.

4. Return all neighbors of x in the graph $G_{s'}$.

End of Algorithm

Let x be a vertex of G. Step 2 determine the values of e, i, j, s such that $x \in L^s_{\langle e,i\rangle}(j) - L^{s-1}_{\langle e,i\rangle}(j)$. Since every time the requirement $R_{\langle e,i\rangle}$ deserves attention, we apply the procedure stated in part 2 of Lemma 10 once and the sequence of finite graphs given by Lemma 10 has finite length, the loop in the algorithm will terminate. It remains to show that $\forall s > s'$, $L^s_{\langle e,i\rangle}(j) = L^{s'}_{\langle e,i\rangle}(j)$. Note that we can exit the loop only when $L^t_{\langle e,i\rangle}(j) = L^{n(t)}_{\langle e,i\rangle}(j)$ and $s' = n(t)$. By the definition of $n(t)$, all the elements in A which are less than or equal to $m(t)$ were enumerated. Hence $\forall s > n(t)$, $L^s_{\langle e,i\rangle}(j)$ does not deserve attention. Hence, $\forall s > s'$, $L^s_{\langle e,i\rangle}(j) = L^{s'}_{\langle e,i\rangle}(j)$. ∎

Lemma 7 $\forall \langle e,i\rangle \geq 0$, $R_{\langle e,i\rangle}$ *is satisfied.*

Proof: We shall show that if $R_{\langle e,i\rangle}$ is not satisfied for some $\langle e,i\rangle$, then for any a we can effectively find a stage s_0 such that

$$A_{s_0} \cap \{0,1,...,a\} = A \cap \{0,1,...,a\}.$$

It follows that A is recursive. Note that e and i are fixed.

Let $\langle e,i\rangle$ be such that $R_{\langle e,i\rangle}$ is not satisfied. Each subgraph $T_{\langle e,i\rangle}(j)$ $(j \geq 0)$ must need attention but does not deserve attention and never will from a certain stage onwards. Let

α_j = number of times $T_{\langle e,i\rangle}(j)$ deserved attention during the course of the construction.

$\beta_{\langle e,i\rangle}$ = maximum $\{k : (\overset{\infty}{\exists} j)[k = \alpha_j]\}$.

$\beta_{\langle e,i\rangle}$ exists because the sequence of finite graphs given by Lemma 10 has finite length. We shall now state a family of algorithms parametrized by h such that when $h = \beta_{\langle e,i\rangle}$, this is an algorithm for A.

Algorithm $ALG(h)$

1. Input a.

2. Run the construction of graph G. Look for a stage s_0 which is large enough so that if

   $m = \max \{ z : z$ is a vertex of $L^{s_0}_{\langle e,i\rangle}(j)$, $L^{s_0}_{\langle e,i\rangle}(j)$ is requiring attention at stage s_0+1, $T_{\langle e,i\rangle}(j)$ has deserved attention h times during stages $s \leq s_0$ and $j \leq s_0 \}$,

   then $m \geq a$.

3. Enumerate A up to the stage $s_0 + 1$. If a is enumerated, then return $a \in A$. Otherwise return $a \notin A$.

End of Algorithm

Since $R_{\langle e,i\rangle}$ is not satisfied, stage $s_0 + 1$ as described in part 2 of the algorithm exists. Let $h = \beta_{\langle e,i\rangle}$. We remark that at stage $s_0 + 1$, the only reason that Bean's lemma does not succeed is because A does not permit it. Formally, it means

$$A_{s_0+1} \cap \{0, 1, ..., m\} = A \cap \{0, 1, ..., m\}.$$

As $m \geq a$, therefore

$$A_{s_0+1} \cap \{0, 1, ..., a\} = A \cap \{0, 1, ..., a\}.$$

Thus algorithm $ALG(\beta_{\langle e,i\rangle})$ is a recursive algorithm for A. ∎

This completes the proof of the main theorem. ∎

Of particular interest are the low sets since they are close to recursive.

Corollary 8 *For any nonrecursive low r.e. set A there exists an A-recursive graph G with $\chi(G) = 2$ and $\chi^r(G) = \infty$.*

4 Conclusions and Further Remarks

We have demonstrated that for any nonrecursive r.e. set A, we can construct an A-recursive graph G such that $\chi(G) = 2$ but $\chi^r(G)$ is not even finite. An obvious open question is to generalize the construction to *any* set A with $\emptyset <_{\mathrm{T}} A <_{\mathrm{T}} \emptyset'$. Note that the construction will be difficult without permitting and we may begin to look at special cases such as when A is a 2-r.e. set ([Ers68a, Ers68b, Ers70, EHK81]).

Another interesting direction is to investigate what kinds of conditions we may impose on recursive graphs so that we can obtain better uniform bounds. We generalize the notion of decidable graphs ([Bea76]) as follows:

Definition 9 Let $G = (V, E)$ be a recursive graph. Define the first order query language $\mathbf{L}$ as the collection of well-formed formulae using the usual logical symbols (including $=$) and

1. constant symbols $0, 1, 2, \ldots, n, \ldots$, intended to represent the vertices.
2. variable symbols $x_0, \ldots, x_n, \ldots$.
3. A single (binary) predicate symbol R, interpreted as the adjacency relation.

Let $f : \mathbf{L} \to \{0, 1\}$ be the function defined as

$$f(\phi) = \begin{cases} 1 & \text{if } \phi \text{ is true,} \\ 0 & \text{if } \phi \text{ is false.} \end{cases}$$

G is said to be A-*decidable* if $f \leq_{\mathrm{T}} A$. When A is recursive, G is said to be *decidable*.

It is shown [Bea76] that there are highly recursive graphs which are not decidable, and for every $k \geq 3$, there exists a k-colorable, decidable graph with no recursive k-colorings. By combining the techniques in [Bea76] and the construction of a highly recursive graph G with $\chi^r(G) = 2\chi(G) - 1$ (see [Sch80, BG89]), one can construct a decidable graph G which satisfies the same equality. It will be of interest to determine the bounds for various sets A and query languages.

Acknowledgement We would like to thank Kalvis Apsitis, Mark Changizi, April Lee and the anonymous referee for proofreading and helpful suggestions.

5 Appendix: Bean's construction

We now state the following technical lemma, which is a variant of Bean's original construction. A similar version of this lemma was stated in [BG89] and we include the proof here for completeness.

Lemma 10

Let $L_0 = (V, E)$ be the graph with 2^i isolated vertices, $\{e\}$ be a Turing machine. Then there exists a finite sequence of finite graphs $L_0, \ldots, L_r$ such that the following conditions hold. (For notation $L_j = (V_j, E_j)$.)

1. *For $1 \leq j \leq r$, L_j is an extension of L_{j-1}.*
2. *For every j, $0 \leq j < r$, L_{j+1} can be obtained recursively from L_j, the values of $\{e\}(x)$ for every $x \in V_j$, and an index of an arbitrary infinite recursive set X. The new vertices introduced to L_{j+1} are all from X and we may assume that if x is a new vertex, then $x > y$ for any vertices $y \in V_j$.*

3. *There exists a nonempty set $W \subseteq V_r$ of vertices such that W witnesses the fact that $\{e\}$ is not an i-coloring of L_r in one of the following way:*

   (a) *$\{e\}$ is not total on W (so $\{e\}$ is not a coloring of L_r and W is said to be a witness of type a), or*

   (b) *there exist $v \in V_r$, $w \in W$ such that $\{v, w\} \in E_r$ and $\{e\}(v) = \{e\}(w)$ (so $\{e\}$ is not a coloring of L_r and W is said to be a witness of type b), or*

   (c) *for all $x \in W$, $\{e\}(x)\downarrow$, and $|\{e\}(W)| = i + 1$ (so $\{e\}$ is not an i-coloring of L_r and W is said to be a witness of type c).*

4. *$\forall i \geq 0$, $\chi(L_i) \leq 2$. Moreover, there is a 2-coloring of L_r in which W is 1-colored.*

5. *$\forall i \geq 0$, L_i is planar.*

Proof: The Turing machine $\{e\}$ is fixed throughout this proof.

We prove this lemma by induction on i. Assume $i = 0$. Let $L_1 = L_r = (\{x\}, \emptyset)$ and $W = \{x\}$. If $\{e\}(x)\uparrow$, then W is a witness of type a. If $\{e\}(x)\downarrow$, then W is a witness of type c. In either case conditions 1–5 are easily seen to be satisfied.

Assume this lemma is true for i. We show it is true for $i + 1$. Let X be an arbitrary infinite recursive set and $X = X_1 \cup X_2$ be a recursive partition of X into two infinite recursive sets such that indices for X_1 and X_2 can be obtained from indices for X. By the induction hypothesis, we obtain the following:

1) a sequence of graphs $L_{11}, L_{21}, \cdots, L_{r_1 1}$, and a set W_1 such that the sequence together with witness set W_1 satisfies 1–5 (note that all the vertices are in X_1), and

2) a sequence of graphs $L_{12}, L_{22}, \cdots, L_{r_2 2}$, and a set W_2 such that the sequence together with witness set W_2 satisfies 1–5 (note that all the vertices are in X_2).

Assume $r_1 \leq r_2$. We define graphs $L_1, L_2, \ldots, L_{r'}$ that satisfy the lemma (r' will be either r_1, r_2 or $r_2 + 1$). For $1 \leq j \leq r_1$ let

$$L_j = L_{j1} \cup L_{j2}.$$

If for all x, x a vertex of $L_{r_1 1}$, $\{e\}(x)\downarrow$, then for $r_1 + 1 \leq j \leq r_2$ let

$$L_j = L_{r_1 1} \cup L_{j2}.$$

(If this does not occur, then L_{r_1} is the final graph and W_1 is the witness set.) In this case we obtain witness sets as follows. If W_1 (W_2) is a witness of type a or b, then L_{r_2} is our final graph and $W = W_1$ (W_2). The 2-coloring of the final graph with the witnesses 1-colored can be obtained by combining such colorings from $L_{r_1 1}$ and $L_{r_2 2}$. It is easy to see that the sequence of graphs and the witness set W all satisfy conditions 1–5.
If both W_1 and W_2 are witnesses of type c, then there are two cases:

(Case 1) If $\{e\}(W_1) \neq \{e\}(W_2)$, then either there is some element $w \in W_1$ such that $\{e\}(w) \notin \{e\}(W_2)$, or there is some element $w \in W_2$ such that $\{e\}(w) \notin \{e\}(W_1)$. We examine the latter case, the former is similar. Our final graph is L_{r_2} and we let $W = W_1 \cup \{w\}$. By the induction hypothesis and the fact that W_1 is of type c, $|\{e\}(W_1)| = i+1$. Since $w \notin W_1$ and $\{e\}(w) \notin \{e\}(W_1)$, $|\{e\}(W_1 \cup \{w\})| = i + 2$. Hence W is a witness of type c. The 2-coloring of the final graph with the witnesses 1-colored can be obtained by combining such colorings from $L_{r_1 1}$ and $L_{r_2 2}$.

(Case 2) If $\{e\}(W_1) = \{e\}(W_2)$, then let w be the least element of X that is bigger than both any element used so far and the number of steps spent on this construction so far (this is done to make the graph recursive). Let

$$\begin{aligned} L_{r_2+1} = &\ L_{r_2} \cup \{\{u, w\} : u \in W_1\} \\ W = &\ W_2 \cup \{w\}. \end{aligned}$$

If $\{e\}(w) \uparrow$, then W is a witness of type a. If $\{e\}(w) \downarrow \in \{e\}(W_1)$, then since w is connected to all vertices in W_1, W is a witness of type b. If $\{e\}(w) \downarrow \notin \{e\}(W_1)$ (and hence $\{e\}(w) \notin \{e\}(W_2)$) then $\{e\}(W) = \{e\}(W_2 \cup \{w\}) = \{e\}(W_2) \cup \{e\}(w)$, which has cardinality $i + 2$; therefore W is a witness of type c. Hence W is a witness set. A 2-coloring of L_{r_2+1} with W 1-colored can easily be obtained from the 2-coloring of $L_{r_1 1}$ (that 1-colors W_1) and the 2-coloring of $L_{r_2 2}$ (that 1-colors W_2).

It is easy to see that the sequence $L_1, L_2, \cdots, L_{r'}$ and the set W satisfy 1–5. ∎

References

[Bea76] D. Bean. Effective coloration. *The Journal of Symbolic Logic*, 41:469–480, 1976.

[BG89] R. Beigel and W.I. Gasarch. On the complexity of finding the chromatic numbers of a recursive graph I: The bounded case. *Annals of Pure and Applied logic*, 45:1–38, 1989.

[CP83] H.G. Carstens and P. Pappinghaus. Recursive colorations of countable graphs. *Annals of Pure and Applied logic*, 25:19–45, 1983.

[EHK81] Richard L. Epstein, Richard Haas, and Richard L. Kramer. Hierarchies of sets and degrees below $\mathbf{0}'$. In *Logic Year 1979–80*, volume 859 of *Lecture Notes in Mathematics*, pages 32–48, Berlin, 1981. Springer-Verlag.

[Ers68a] Yu. L. Ershov. A hierarchy of sets, I. *Algebra i Logika*, 7(1):47–74, January–February 1968. English Translation, Consultants Bureau, NY, pp. 25–43.

[Ers68b] Yu. L. Ershov. A hierarchy of sets, II. *Algebra i Logika*, 7(4):15–47, July–August 1968. English Translation, Consultants Bureau, NY, pp. 212–232.

[Ers70] Yu. L. Ershov. A hierarchy of sets, III. *Algebra i Logika*, 9(1):34–51, January–February 1970. English Translation, Consultants Bureau, NY, pp. 20–31.

[Sch80] J. Schmerl. Recursive coloring of graphs. *Canadian Journal of Mathematics*, 32:821–830, 1980.

[Soa87] R.I. Soare. *Recursively Enumerable Sets and Degrees*. Omega Series. Springer-Verlag, 1987.