Annals of Pure and Applied Logic 167 (2016) 235–246

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

A-computable graphs

Matthew Jura [a], Oscar Levin [b,*], Tyler Markkanen [c]

[a] Department of Mathematics, Manhattan College, 4513 Manhattan College Parkway, Riverdale, NY 10471, USA
[b] School of Mathematical Sciences, University of Northern Colorado, Greeley, CO 80639, USA
[c] Department of Mathematics, Physics, and Computer Science, Springfield College, 263 Alden Street, Springfield, MA 01109, USA

ARTICLE INFO

Article history:
Received 29 August 2015
Received in revised form 18 October 2015
Accepted 6 November 2015
Available online 21 November 2015

MSC:
03D25
03D28
03D45
03C57

Keywords:
Computability theory
Euler paths
Chromatic number
Highly computable graphs

ABSTRACT

We consider locally finite graphs with vertex set $\mathbb{N}$. A graph G is *computable* if the edge set is computable and *highly computable* if the neighborhood function N_G (which given v outputs all of its adjacent vertices) is computable. Let $\chi(G)$ be the chromatic number of G and $\chi^c(G)$ be the computable chromatic number of G. Bean showed there is a computable graph G with $\chi(G) = 3$ and $\chi^c(G) = \infty$, but if G is highly computable then $\chi^c(G) \leq 2\chi(G)$. In a computable graph the neighborhood function is Δ^0_2. In highly computable graphs it is computable. It is natural to ask what happens between these extremes. A computable graph G is A-*computable* if $N_G \leq_T A$. Gasarch and Lee showed that if A is c.e. and not computable then there exists an A-computable graph G such that $\chi(G) = 2$ but $\chi^c(G) = \infty$. Hence for A noncomputable and c.e., A-computable graphs behave more like computable graphs than highly computable graphs. We prove analogous results for Euler paths and domatic partitions. Gasarch and Lee left open what happens for other Δ^0_2 sets A. We show that there exists an $\emptyset <_T A <_T \emptyset'$ such that every A-computable graph G with $\chi(G) < \infty$ has $\chi^c(G) < \infty$. Finally, we classify all such A.

1. Introduction

When applying notions in computability theory to results in graph theory, we often find that classical results are not effectively true. Two early examples of this phenomenon are due to Bean: there is a computable graph with an Euler path but no computable Euler path [2], and there is a computable graph with chromatic number 3 but no computable finite coloring [1] (in fact there is a computable graph with chromatic number 2 with no computable finite coloring if you don't require the graph to be connected [13]). Results of this flavor abound including ones for Hamilton paths [5], perfect matchings [8], edge colorings [7], and domatic partitions [6].

* Corresponding author.
E-mail addresses: matthew.jura@manhattan.edu (M. Jura), oscar.levin@unco.edu (O. Levin), tmarkkanen@springfieldcollege.edu (T. Markkanen).

http://dx.doi.org/10.1016/j.apal.2015.11.003
0168-0072/

Some of these results (including the original two of Bean) rely heavily on our ability to add neighbors to a given vertex in the construction of the graphs. A *computable graph* is simply a graph for which the edge relation is computable (here and throughout the paper we assume without loss of generality that computable graphs have vertex set $\mathbb{N}$). Thus there is an effective procedure to decide whether two given vertices are in fact adjacent. However, we cannot in general produce the list of all vertices adjacent to a given vertex (or equivalently, determine the degree or valency of a given vertex). Those computable graphs which have a computable neighborhood function, which, when given a vertex, outputs the list of all vertices adjacent to it (i.e., its neighbors), are called *highly computable*. For this definition to make sense, all vertices must have only finitely many neighbors (such graphs are called *locally finite*), and in this paper we will only consider such graphs.

Whenever a construction of a graph requires adding neighbors to vertices arbitrarily late in the construction, we wonder whether this requirement is necessary. Often it is, in that the result that holds for computable graphs fails to hold for highly computable graphs. Indeed, every highly computable graph with chromatic number n has a computable $(2n-1)$-coloring (proved independently by Schmerl [13] and Carstens and Päppinghaus [3]; Schmerl proved this bound is tight). Similarly, every highly computable graph containing an Euler path has a computable Euler path [2].

The goal of this paper is to better understand this behavior by investigating graphs that are *between* computable and highly computable. We adopt the approach suggested by Gasarch and Lee [4] and consider *A-computable graphs*[1] for various Δ^0_2 sets A, meaning A computes the neighborhood function. We denote the neighborhood function of G by N_G. Specifically, given a vertex v of G, $N_G(v)$ returns the canonical index for the finite set of vertices that are adjacent to v in G.

Definition 1.1 *(Gasarch and Lee).* Let A be a set. A locally finite graph $G = (V, E)$ is *A-computable* provided G is computable and $N_G \leq_T A$.

Note that computable graphs always have a neighborhood function computable from the halting problem K, so they are K-computable, while highly computable graphs have computable neighborhood function, making them $\emptyset$-computable.

Since A-computable graphs might be somewhere between computable and highly computable, it is reasonable to wonder whether the complexity of graph-theoretic properties of A-computable graphs might be between those of computable and highly computable graphs. Gasarch and Lee considered this question for vertex colorings and found that at least for noncomputable c.e. sets A, the A-computable graphs behave just like the computable ones. The following is the main result of their paper.

Theorem 1.2 *(Gasarch and Lee). (See [4].) Let A be a noncomputable c.e. set. There exists an A-computable graph G such that G is 2-colorable but not computably k-colorable for any natural number k.*

The authors employed the technique of c.e.-permitting in their construction. They asked:

Question 1.3. Can Theorem 1.2 be extended to any set A with $\emptyset <_T A <_T \emptyset'$?

They remarked that the construction would be more difficult without permitting, and suggested that it might be easier first to consider the case when A is 2-c.e. Indeed, there is a version of permitting with Δ^0_2 sets, aptly called Δ^0_2-permitting. (See [9] for a nice exposition of this method.) Unfortunately, the method of Δ^0_2-permitting does not seem to apply when constructing an A-computable graph. The reason is that, while the neighborhood function of an A-computable graph is necessarily Δ^0_2, we cannot remove an edge from an

[1] Gasarch and Lee used the term *A-recursive* to denote the same concept.