Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 345 (2005) 331–344
Theoretical Computer Science
www.elsevier.com/locate/tcs

Graph exploration by a finite automaton☆

Pierre Fraigniaud[a,1], David Ilcinkas[a,1], Guy Peer[b], Andrzej Pelc[c,2], David Peleg[b]

[a]*CNRS, LRI, Université Paris-Sud, France*
[b]*Department of Computer Science, Weizmann Institute, Israel*
[c]*Dépt. of d'informatique, Univ. du Québec en Outaouais, Canada*

Abstract

A finite automaton, simply referred to as a *robot*, has to explore a graph whose nodes are unlabeled and whose edge ports are locally labeled at each node. The robot has no a priori knowledge of the topology of the graph or of its size. Its task is to traverse all the edges of the graph. We first show that, for any K-state robot and any $d \geqslant 3$, there exists a planar graph of maximum degree d with at most $K+1$ nodes that the robot cannot explore. This bound improves all previous bounds in the literature. More interestingly, we show that, in order to explore all graphs of diameter D and maximum degree d, a robot needs $\Omega(D \log d)$ memory bits, even if we restrict the exploration to planar graphs. This latter bound is tight. Indeed, a simple DFS up to depth $D+1$ enables a robot to explore any graph of diameter D and maximum degree d using a memory of size $O(D \log d)$ bits. We thus prove that the worst case space complexity of graph exploration is $\Theta(D \log d)$ bits.

Keywords: Graph exploration; Labyrinth; Finite automaton; Mobile agent; Robot

☆ A preliminary version of this paper appeared in the Proceedings of the 29th International Symposium on Mathematical Foundations of Computer Science (MFCS) [29].

E-mail addresses: (P. Fraigniaud), ilcinkas@lri.fr (D. Ilcinkas), guy850@zahav.net.il (G. Peer), pelc@uqo.ca (A. Pelc), david.peleg@weizmann.ac.il (D. Peleg).

[1] Supported by the project "PairAPair" of the ACI Masses de Données, the project "Fragile" of the ACI Sécurité et Informatique, and by the project "Grand Large" of INRIA.

[2] Supported in part by NSERC Grant OGP 0008136 and by the Research Chair in Distributed Computing of the Université du Québec en Outaouais.

0304-3975/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2005.07.014

1. Introduction

1.1. Background and motivation

A mobile entity, e.g., a software agent or a robot, has to *explore* an undirected graph by visiting all its nodes and traversing all its edges, without any a priori knowledge of the topology of the graph or of its size. The task of visiting all nodes is fundamental in searching for data stored at unknown nodes of a network, and traversing all edges is often required in network maintenance and when looking for defective components. More precisely, we consider the task of "perpetual" exploration in which the robot has to traverse all edges of the graph but is not required to stop. That is, the robot moves from node to node, traversing edges, so that eventually all edges have been traversed. Perpetual exploration is of practical interest, e.g., if regular control of a network for the presence of faults is required, and all edges must be periodically traversed over long periods of time.

If nodes and edges have unique labels, exploration can be easily achieved (e.g., by depth-first search). However, in some navigation problems in unknown environments, such unique labeling may not be available, or limited sensory capabilities of the robot may prevent it from perceiving such labels. Hence, it is important to be able to program the robot to explore *anonymous* graphs, i.e., graphs without unique labeling of nodes or edges. Clearly, the robot has to be able to locally distinguish ports at a node: otherwise it is impossible to explore even the star with 3 leaves (after visiting the second leaf, the robot cannot distinguish the port leading to the first visited leaf from that leading to the unvisited one). Hence, we make a natural assumption that all ports at a node are locally labeled $1, \ldots, d$, where d is the degree of the node. No consistency between those local labelings is assumed.

In many applications, robots and mobile agents are meant to be simple, often small and inexpensive devices. This limits the amount of memory with which they can be equipped. As opposed to numerous papers that imposed no restrictions on the memory of the robot and sought exploration algorithms minimizing time, i.e., the number of edge traversals, we investigate the minimum memory size of the robot that allows exploration of graphs of given (unknown) size, regardless of the time of exploration. That is, we want to find an algorithm for a robot performing exploration, using as little memory as possible.

A robot with a k-bit memory is modeled as a finite automaton. The first known finite automaton algorithm designed for graph exploration was introduced by Shannon [46] in 1951. Since then several papers have been dedicated to the graph exploration problem. In 1967, during his talk at Berkeley, Rabin [43] proposed a conjecture that no finite automaton with a finite number of pebbles can explore all graphs (a pebble is a marker that can be dropped at and removed from nodes). In 1971, Müller [39] gave some formal arguments to support Rabin's claim, in the restricted case of a robot without pebbles. In 1977, Coy [20] presented another proof, but some parts of it are fuzzy. The first formal proof of Rabin's claim is generally attributed to Budach [18], in 1978, for a robot without pebbles. Actually, the long and technical paper by Budach is concerned with labyrinths. A *labyrinth* is a two-dimensional obstructed chess-board (i.e., $\mathbb{Z}^2$ with forbidden cells). The forbidden cells in $\mathbb{Z}^2$ are described by a set L. If L (resp., $\mathbb{Z}^2 \setminus L$) is finite, then the labyrinth is called