DFS TREE CONSTRUCTION: ALGORITHMS and CHARACTERIZATIONS

(Preliminary version)

Ephraim Korach and Zvi Ostfeld

Computer Science Department
Technion - Israel Institute of Technology
Haifa 32000, Israel

ABSTRACT

The Depth First Search (DFS) algorithm is one of the basic techniques which is used in a very large variety of graph algorithms. Every application of the DFS involves, beside traversing the graph, constructing a special structured tree, called a *DFS tree*. In this paper, we give a complete characterization of all the graphs in which every spanning tree is a DFS tree. These graphs are called *Total–DFS–Graphs*. The characterization we present shows that a large variety of graphs are not *Total–DFS–Graphs*, and therefore the following question is naturally raised: *Given an undirected graph* $G=(V,E)$ *and an undirected spanning tree T, is T a DFS tree of G?* We give an algorithm to answer this question in linear ($O(|E|)$) time.

1. INTRODUCTION

The Depth First Search (DFS) algorithm is one of the basic techniques which is used in a very large variety of graph algorithms. The history of this algorithm (in a different form) goes back to 1882 when Tremaux' algorithm for the maze problem was first published (see [BLW, page 18]).

The impact of *DFS* grew rapidly since the Hopcroft and Tarjan version of it was published (see [Ta], [HT a], [HT b], and [HT c]).

In many areas of computer science, this algorithm is used, and lately it also has penetrated the field of parallel and distributed algorithms (e.g. [AA], [Aw], [KO], [LMT], [Re], and [Ti]).

Every use of the *DFS*, beside traversing the graph, constructs a special structured directed rooted tree, called a *DFS tree*, that may be used subsequently.

In this paper, we raise two important questions, regarding the structure of the *DFS* tree that is obtained, and discuss their solutions.

(i) First, we are interested in determining in which graphs, every spanning tree can be obtained as a *DFS* tree. These graphs are called *Total–DFS–Graphs* (*T–DFS–G*). In section 2 we give a complete characterization of these graphs.

It turns out that a large variety of graphs are not *Total–DFS–Graphs*, and therefore the following question is naturally raised:

(ii) *Given an undirected graph $G=(V,E)$ and an undirected spanning tree T of G, is T a DFS tree (T-DFS) in G?*

Question (ii) is answered by a linear ($O(|E|)$) time algorithm in section 3. The algorithm gives as an output all the vertices $S = \{ s \in V : T_s$ - the directed tree rooted at s - is a *DFS* tree in G $\}$. If T is not a *DFS* tree in G (i.e. S is the empty set) then the algorithm can supply an $O(|V|)$ proof for that fact. Details about that feature of the algorithm will appear in the final version of this paper.

One can think about many applications of the results presented here. For example, we can constructively solve the following problem: Let G be an undirected graph with a unique minimum undirected spanning tree T (e.g. where no two edges have the same weight). Is it possible to run a *DFS* in such a way that T will be the *DFS* tree ? (This question might be