Construction of AND – OR subgraph of a DAG

Thinh D. Nguyen
Moscow State University
kosmofarmer@gmail.com

Abstract. *The aim of this paper is to provide some deviating approaching method to create mathematical claims with Sharygin’s pencil drawings. Readers are encouraged to read this article with one piece of paper and a pencil to note whatever is intuitive to understand what follows.*

Keywords: *sample, paper*

1. INTRODUCTION

One can find many useful sources of mathematical papers online related to the theory of computing. Bound to that same theory, the main aim of this paper is to prove a small result on a computational graph theory problem. It is not the aim of this paper to show how a particular mathematical discipline would shape the whole mathematics as a pioneering modern science. In this section, we give the notation and definitions commonly seen in any graph theory monograph. We give the definition of our particular problem at the end of this section.

1.1. Graph theory notions. A *digraph* $G = (V, E)$ is a directed graph with the *vertex set* $V = V(G)$ and *edge* (arc, in some monographs) *set*. In a digraph, every edge (u, v) ($u, v \in V$) is directed, in the sense that it has u as its starting vertex and v as its ending vertex. A digraph is called a *DAG* (Directed Acyclic Graph) whenever it contains no *directed cycle*. And, a directed cycle in a digraph G is a sequence of vertices $(v_1, v_2, \dots, v_k)$ in which, for all $1 \leq i < k$, (v_i, v_{i+1}) is an edge of G , and also (v_k, v_1) is an edge of G . These are all the definitions and notations from graph theory that are needed in this paper.

1.2. Labelled DAG. Now, we give the definitions that are pertinent to our soon-to-be-defined problem. An AND – OR DAG is one which vertices are all labelled with either AND or OR labels. The meanings of these two labels will become natural after one comprehends the notion of a *feasible* subgraph of an AND – OR DAG which is described as follows. First, every vertex in a DAG is either *terminal* or *non-terminal* one. A *terminal* is a vertex without outgoing edges. A vertex that is not a terminal is a *non-terminal*. Second, vertex v is a *child* of vertex u when and only when (u, v) is an edge of the DAG. So, the *children* of a vertex u are all the vertices that outgoing edges from u point to. Third, having all that clearly defined, we now define a *feasible subgraph* G' of an AND – OR DAG G as one that satisfies the following:

- G' has a *root* $s \in V(G')$, where a root of a DAG is the vertex from which one can reach every other vertex of that DAG
- If $v \in V(G')$ and v is an AND vertex, then all of v ’s children in G are also present in G'
- If $v \in V(G')$ and v is an OR vertex, then some of v ’s children in G is also present in G'

1.3. Definition and main claim. Now, we give the definition of our problem.

Definition 1.1. : AND – OR DAG SUBGRAPH = $\{ \langle G, s, k \rangle \mid G \text{ which is an AND – OR DAG contains a feasible subgraph } G' \text{ which has no more than } k \text{ vertices and has } s \text{ as its root} \}$

We close this section by our main observation.

Claim 1.2. $\text{SAT} \leq_p \text{AND – OR DAG SUBGRAPH}$

2. PROOF OF CLAIM 1.2

In this section we present the proof of claim 1.2.

Proof. As defined in definition 1.1, an AND – OR DAG $\mathbb{G}$ together with a source vertex s and an integer k can be an instance of our problem. We use VERTEXCOVER as a hard problem to reduce from. So, we are given an instance of VERTEXCOVER consisting of a graph G and integer k as input.

Our $\mathbb{G}$ is an AND – OR DAG which terminals represent the vertices of G and which OR vertices represent the edges also of G . From an OR vertex $e = (u, v) \in E(G)$ in $\mathbb{G}$, we draw two edges to its starting and ending vertices u, v in G and as G -vertex-vertices also vertices of $\mathbb{G}$. Finally, to finish our construction, we add the root vertex s (that will also be given in the produced AND – OR DAG SUBGRAPH instance) as the only AND vertex that has one outgoing edge to each OR vertex.

Clearly, G has a k -vertex cover if and only if $\mathbb{G}$ has some k -vertex feasible subgraph $\mathbb{G}'$ with root s . The claim does hold therefore. $\square$

2.1. Some additional observations. One can continue to extend the above constructions to many other computational problems keeping in mind that mathematical constructions are only beautiful when they are inclined towards fruitful landscapes where one can find one's self happy and dreamful. We would like to thank all the Soviet-era mathematical writers for all their pages of dreamful theorems, lemmas, corollaries, etc. All these pages that have immense impacts on future mathematics will remain forever with humanity.

3. CONCLUSION

From time to time, one needs to ask one's self whether mathematics in this century is really improving ever since the previous one. Practically, one is gotten used to tapping into Internet regularly, for at least over the few last decades, and sometimes one really finds novel maths tricks which one manages to nourish into one's own theorems. We emphasize again that if one has ever loved reading maths books of Sharygin, one would question the capability of some day proving some mathematical conjecture.

3.1. Concluding gallery. We would like to introduce the readers to some relaxing pictures of front cover of maths books of I. F. Sharygin

REFERENCES

- [1] Michael R. Garey, David S. Johnson, **Computers and Intractability: A Guide to the Theory of NP-completeness**
- [2] The free encyclopedia Wikipedia, <https://en.wikipedia.org/wiki/NP-completeness>