

Examining Classical Graph-Theory Problems From the Viewpoint of Formal-Verification Methods (Invited Talk)

Orna Kupferman

orna@cs.huji.ac.il

The Hebrew University

School of Computer Science and Engineering

Jerusalem 91904, Israel

ACM Reference format:

Orna Kupferman. 2017. Examining Classical Graph-Theory Problems From the Viewpoint of Formal-Verification Methods (Invited Talk). In *Proceedings of 49th Annual ACM SIGACT Symposium on the Theory of Computing, Montreal, Canada, June 2017 (STOC'17-KEY)*, 1 pages.

DOI: 10.1145/3055399.3079075

structure into account, and another goal is to reason about a succinct representation of the graph without constructing its flat expansion.

1 INTRODUCTION

While classical graph algorithms are extensively studied by the “Volume A” community of Theoretical Computer Science, the rich semantics and structure that graphs may have is the bread and butter of the “Volume B” community. In particular, in formal verification, we study labeled and game graphs, and we develop algorithms for systems with an internal structure or with huge state spaces. The questions studied by the formal-verification community are, however, very different from those studied in classical graph theory.

The talk surveys a series of works that lift the rich semantics and structure of graphs, and our experience in reasoning about them, to classical graph-theoretical problems. For example, in *labeled graphs*, each edge or vertex is labeled by a letter from some alphabet, describing an action (e.g., in planning or in VLSI design), a query (e.g., in databases), a property like its provider or its security level (e.g., in a network of channels), and many more. The input to the problems includes a language over the alphabet that serves as a specification to paths found by the algorithm. Then, searching for shortest, Eulerian, or Hamiltonian paths, one may restrict attention to paths satisfying the specification, and similarly for paths in a max-flow network along which flow is streamed, or paths in a communication network along which messages are transmitted. As another example, in *game graphs*, vertices and edges may be controlled by different entities (e.g., in the modeling of reactive or multi-agent systems, where transitions depend on actions taken by the underlying entities), and the algorithms have control only in a subset of the vertices. Then, in a max-flow game, we can direct the flow only in some of the vertices and we seek a strategy maximizing the flow against a hostile direction of flow in vertices we do not control. Finally, reasoning about a graph with an *internal structure*, one goal is to enrich the problem definition in a way taking the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

STOC'17-KEY, Montreal, Canada

© 2017 ACM. 978-1-4503-4528-6/17/06...\$15.00

DOI: 10.1145/3055399.3079075