Generating Graphs Uniformly at Random in User-Specified Domains

PhD Project Proposal

Detlef Plump

This project will contribute to the departmental research theme Critical Systems in that it develops a method for generating (black-box) test data with a uniform distribution guarantee for programs working on graph-like structures. This includes test data for classical graph algorithms but also, for example, pointer-data structures for testing C programs. The uniform distribution ensures that the space of possible inputs for a critical algorithm is fairly covered, independently of the number of test graphs required by the user.

The hallmark of this project will be that generated graphs are uniformly distributed in a *user-specified* domain with properties relevant to an application. For example, the application domain may be restricted to acyclic graphs, certain forms of trees, or pointer-data structures such as cyclic lists.

Available graph generators lack the highly desirable feature of generating graphs within an application-specific domain. But there exist random string generators which generate strings uniformly at random within a user-defined context-free language. The goal of this project is to develop algorithms that generate graphs uniformly at random within a user-defined graph class. The graph class may be defined by a context-free graph grammar [3], a graph reduction system or even by a graph program [5].

In the undergraduate project [2], a tool has been implemented which produces random graphs over context-free graph grammars by adapting the method of [4] from strings to graphs. The user specifies a hyperedge-replacement grammar [3] together with a size range, and the tool generates a sequence of random graphs from the language of that grammar. The point of this generator is that users can specify the shape of the graphs to be generated by means of the graph grammar. For example, one may want to generate flow graphs or logical circuits only.

The applicability of [2] is limited in that it requires an unambiguous input grammar in order to achieve a uniform distribution. As practical examples of graph grammars are often ambiguous, many isomorphic copies of already generated graphs may be produced for such grammars. In contrast, the approach of [6, 1] generates strings uniformly at random over possibly ambiguous context-free string grammars. The goal of this PhD project is to adapt this method to context-free graph grammars and other graph generating formalisms.

In addition, the generation algorithm should be generalized to generate graphs randomly with some selected non-uniform distributions (such as the Poisson distribution) which are suitable for certain applications.

References

[1] A. Bertoni, M. Goldwurm, and M. Santini. Random generation and approximate counting of ambiguously described combinatorial structures. In *Proc. 17th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2000)*, volume 1770 of *Lecture Notes in Computer Science*, pages 567–580. Springer, 2000.

[2] J. Coxon. Uniform random generation of graphs with graph grammars. BEng dissertation, University of York, 2013.

[3] F. Drewes, A. Habel, and H.-J. Kreowski. Hyperedge replacement graph grammars. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation. Volume I: Foundations*, chapter 2, pages 95–162. World Scientific, 1997.

[4] T. J. Hickey and J. Cohen. Uniform random generation of strings in a context-free language. *SIAM Journal on Computing*, 12(4):645–655, 1983.

[5] D. Plump. The graph programming language GP. In *Proc. International Conference on Algebraic Informatics (CAI 2009)*, volume 5725 of *Lecture Notes in Computer Science*, pages 99–122. Springer, 2009.

[6] M. Santini. *Random Generation and Approximate Counting of Combinatorial Structures.* PhD thesis, Computer Science Department, University of Milan, 2000. Published as CoRR abs/1012.3000, 2010.