A Mechanized Proof of Tarjan's Algorithm in the TLA$^+$ Proof System

General information

Supervisor	Stephan Merz
Adress	Inria Nancy – Grand Est, Villers-lès-Nancy
Phone	+33 3 54 95 84 78
Email	stephan.merz@loria.fr

Context

Two vertices in a graph are *strongly connected* if they are reachable from each other by following edges of the graph. A set S of vertices is a *strongly connected component* of the graph if all vertices in S are strongly connected and if no superset $T \supseteq S$ of vertices has this property. Computing strongly connected components of a graph is a fundamental algorithmic problem that has applications in temporal logic model checking, among others. Tarjan [7] introduced a linear-time algorithm for computing strong components in a single depth-first pass over the graph. In a recent paper [2], we presented a formulation of Tarjan's algorithm as a functional program, together with formal correctness proofs carried out in the proof assistants Why3 [1], Coq [3], and Isabelle [6], and we invited users of other proof assistants to contribute proofs of the algorithm in their system, with the objective of comparing the strengths and weaknesses of different proof assistants on a non-trivial but manageable case study.

Internship subject

The objective of this internship proposal is to formalize Tarjan's algorithm in the specification language TLA$^+$ [5], and to verify its correctness using TLAPS, the TLA$^+$ Proof System [4] that is in part developed in our research group. The formalization will be closely aligned with the presentation of the algorithm in [2], but it will be based on a transition system rather than two mutually recursive functions. The correctness argument will adapt the invariants given in [2] to the TLA$^+$ specification.

The student working on this subject should be interested in mechanized reasoning about algorithms. No previous knowledge of graph algorithms or of TLA$^+$ is required. A first step of the internship will consist in gaining familiarity with TLA$^+$ specifications and TLAPS proofs.

Work environment

The TLA$^+$ Toolbox, based on Eclipse, provides an IDE for editing TLA$^+$ specifications and for interacting with the proof system. It is readily available for Windows, Linux, and MacOS operating systems.

The internship will take place in Nancy within the VeriDis team of Inria Nancy – Grand Est, a stimulating international research group that is common to Inria, CNRS, University of Lorraine, and the Max-Planck Institute for Informatics in Saarbrücken, and is located at LORIA on the science campus of Nancy. The city of Nancy is a lively university town of intermediate size (about 300,000 inhabitants) in the North-East of France. It offers affordable housing and is home to a rich cultural life and historic treasures, in particular from the 18th and the early 20th century.

References

[1] François Bobot, Jean-Christophe Filliâtre, Claude Marché, Guillaume Melquiond, Andrei Paskevich. The Why3 platform, version 0.86.1. Technical Report, LRI, CNRS, Univ. Paris Sud, Inria Saclay, May 2015. http://toccata.lri.fr/gallery/why3.en.html.

[2] Ran Chen, Cyril Cohen, Jean-Jacques Lévy, Stephan Merz, Laurent Théry. Formal Proofs of Tarjan's Strongly Connected Components Algorithm in Why3, Coq and Isabelle. 10th Intl. Conf. Interactive Theorem Proving (ITP). Leibniz Intl. Proc. in Informatics, 2019. http://drops.dagstuhl.de/opus/volltexte/2019/11068/

[3] The Coq Development Team. The Coq Proof Assistant v.8.3. Reference Manual. Inria, 2010. http://coq.inria.fr/

[4] TLA$^+$ Proofs. Denis Cousineau, Damien Doligez, Leslie Lamport, Stephan Merz, Daniel Ricketts, Hernán Vanzetto. 18th Intl. Symp. Formal Methods (FM). Springer LNCS 7436, pp. 147-154. Paris, France, 2012.

[5] Leslie Lamport. Specifying Systems. Addison Wesley (Boston, Mass.), 2002. http://lamport.azurewebsites.net/tla/tla.html.

[6] Tobias Nipkow, Lawrence Paulson, Markus Wenzel. Isabelle/HOL. A Proof Assistant for Higher-Order Logic. Springer LNCS 2283, 2002.

[7] Robert Tarjan. Depth first search and linear graph algorithms. SIAM Journal on Computing, 1972.