

PROGRAM ANALYSIS, ALGORITHMS & FORMAL METHODS

Faculty: Ganesh Gopalakrishnan, Mike Kirby, John Regehr, Claudio Silva, Konrad Slind, Suresh Venkatasubramanian

Formal Methods and Verification

Research on formal methods in the School of Computing is unique in its tight integration with systems research activities at Utah and elsewhere.

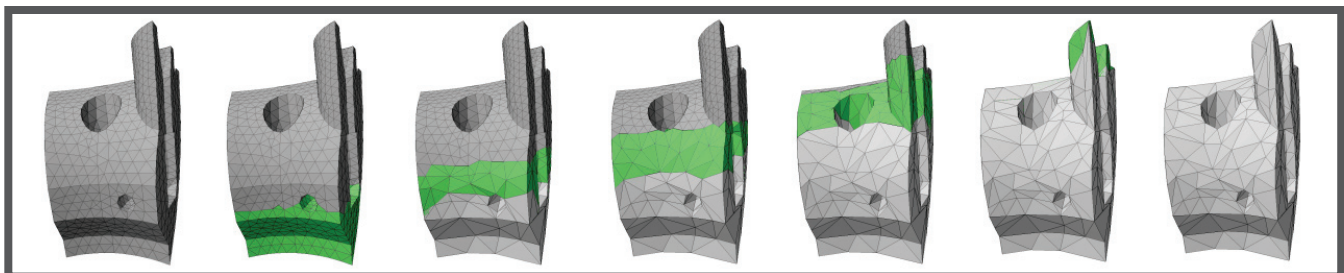
In one research thrust, Profs. Gopalakrishnan and Kirby are collaborating to verify the correctness of communication structures in large-scale aggressively optimized parallel simulations written using the Message Passing Interface (MPI). Another effort led by Prof. Gopalakrishnan is aimed at improving the reliability of hardware cache coherence protocols through formal verification.

Prof. Slind is currently pursuing research on compiler correctness, with the goal of building validating compilers for security-critical applications. Such compilers formally prove the correctness of each successful run of the compiler. Security infrastructure, such as block ciphers and Elliptic Curve Cryptography are currently being used as examples. Prof. Slind also leads the development of the HOL-4 proof assistant, which is used in hardware and software verification.

Prof. Regehr's group builds tools that use lightweight formal methods to verify novel properties of embedded software. For example one tool uses abstract interpretation to bound the stack memory consumption of a compiled sensor network application. This is difficult because real embedded software uses many idioms that are hard to analyze, such as interrupts, recursion, and function pointers.

Algorithms

Algorithms research at the School of Computing explores problems in numerous areas, including topics in massive data sets, data mining, computational geometry, shape analysis and data visualization. One strand of current research deals with the computational challenges of doing statistics on large data sets, and how information-theoretic methods can be brought to bear on a variety of problems in data management. Another thread studies the geometric underpinnings of distributions, and algorithmic questions that arise in this context. Our interest in large data sets leads to basic questions about how to model computations on such data, which connects with work on stream architectures and processors in other disciplines.



Streaming simplification performed on a tetrahedral mesh ordered from bottom to top. The portion of the mesh that is in-core at each step is shown in green