

Developing Disciplined Programs

Seminar at Appalachian State University

Clément Aubert

06/02/2017

Abstract

It happens to every one of us to see a program “freeze”, and we may sometimes notice that a program occupy more memory than what we actually had available. While those are minor downsides in most of our daily usage, it can lead to dramatic consequences when human lives, security, or spatial exploration are at stakes. Most procedures test programs once they have been written and compiled, and provide an a posteriori seal that the program will behave according to its specification. My research area designs programming languages that can only produce “pre-certified” programs, i.e., programs that already come with that seal. The a posteriori test was replaced by the mathematical theorem guaranteeing that the program will end within pre-computed bounds on memory and time.

In this talk, I will sketch the goals and benefits of this approach, called Implicit Computational Complexity. One of my contribution was to identify simple principles in logic programs that can guarantee bounds on time or space. It uses proof-, automata-, and type-theory, a bit of complexity, but reached a stage of maturity that allow to express those results in simple terms, and to develop fascinating perspectives.

When and Where?

When

Monday, February 6, 2017, 2:30 PM – 3:30 PM

Where

Appalachian State University, Anne Belk Hall, Room 318

Resources

Slides

[Here they are!](#)

Research Papers

All my publications are available at <https://lacl.fr/~caubert/#publications>, at [arXiv](#) and at [HAL](#).

Misc.

- Download [a pdf version](#) or [the md source](#) of this page
- [HTML5](#) and [CSS3](#) valid
- [Creative Commons Attribution 4.0 International License](#)
- Contact: auberc@appstate.edu
- Created with [debian](#), [pandoc](#), [latex](#) and [emacs](#).