



THE WEIZMANN INSTITUTE OF SCIENCE
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Guest Seminar

Room 1 ,Ziskind Building
on Thursday, Dec 08, 2016
at 12:00

Shachar Itzhaky
MIT

From Programming Languages to Programming Systems - Software Development
by Refinement

Abstract:

Everyone wants to program with "high-level concepts", rather than meddle with the fine details of the implementation, such as pointers, network packets, and asynchronous callbacks. This is usually achieved by introducing layers of abstraction - but every layer incurs some overhead, and when they accumulate, this overhead becomes significant and sometimes prohibitive. Optimizing the program often requires to break abstractions, which leads to suboptimal design, higher maintenance costs, and subtle hard-to-trace bugs.

I will present two recent projects that attempt to address this difficulty. STNG is an automated lifting compiler that can synthesize high-level graphics code for computing stencils over matrices, from low-level legacy code written in Fortran. Its output is expressed in Halide, a domain-specific language for image processing that can take advantage of modern GPUs. The result is therefore code that is both easier to understand and more efficient than the original code.

Bellmania is a specialized tool for accelerating dynamic-programming algorithms by generating recursive divide-and-conquer implementations of them. Recursive divide-and-conquer is an algorithmic technique that was developed to obtain better memory locality properties. Bellmania includes a high-level language for specifying dynamic programming algorithms and a calculus that facilitates gradual transformation of these specifications into efficient implementations. These transformations formalize the divide-and-conquer technique; a visualization interface helps users to interactively guide the process, while an SMT-based back-end verifies each step and takes care of low-level reasoning required for parallelism.

The lesson is that synthesis techniques are becoming mature enough to play a role in the design and implementation of realistic software systems, by combining the elegance of abstractions with the performance gained by optimizing and tuning the fine details.