Previous Research Experience

Marie-Christine R. Chevalier

My first extended research experience was in summer 1999, in the Computer Science department at the University of California, Berkeley. I was a participant in the SUPERB REU program (Summer Undergraduate Program in Engineering Resarch at Berkeley). I worked with Sanjoy Dasgupta (a graduate student at Berkeley at the time), and Professor Umesh Vazirani. My goal was to implement an algorithm for learning Gaussian mixtures, which was developed by Dasgupta. Dasgupta theorized that this algorithm was simpler than but more effective than the established algorithm for this problem (known as the EM algorithm) [2]. Working independently, I implemented the algorithm in C and experimented with running it on sample data sets; by doing so, I determined that the algorithm's performance could be improved by using random sampling to choose the initial guesses for means, rather than iterating over the entire data set.

More inspiring to me has been my participation in the Lumberjack Project at Wellesley College. The project, headed by Franklyn Turbak in the Computer Science department at Wellesley, is ongoing and has involved several Wellesley students in research on deforestation. Deforestation is a program transformation whose goal is automatically remove virtual data structures from programs. In a given program, virtual data structures are those intermediate data structures which are not necessary to perform the computation at hand, but which are merely created in order to maintain the modular style of the program. In functional programming languages, programs are usually written in a highly modular style with many virtual data structures, so deforestation can potentially reduce the inefficiency associated with functional languages without curtailing any of their benefits. During the 1990s, several algorithms for deforestation were developed, such as warm fusion, but all of them rely on assumptions about the form of the input program ([5], [3], [4]).

I began my involvement in the Lumberjack Project in spring 1999, when I wrote a desugarer for VAL, a language which students in the project were using to experiment with warm fusion. In summer 2000, I participated in the Lumberjack Summer Camp, which was directed by Professor Turbak and by Patricia Johann from the CS department at Bates College and brought together students from Wellesley and Bates to evaluate and compare various approaches to deforestation. During the summer, I primarily worked independently on my project, but frequently discussed my own and others' work with the other students in the program. I began to experiment with an implementation of type-inference-based deforestation, and am currently continuing the work as my senior thesis project.

The type inference algorithm for deforestation, recently developed by Olaf Chitil, is substantially different from other deforestation methods: it performs deforestation using type information, which compilers for functional languages normally derive anyway [1]. In theory, the type inference algorithm seems to be more flexible than previous methods, but the algorithm has not been fully implemented. Over the summer, I spent time familiarizing myself with Chitil's incomplete prototype of the type inference algorithm, and then began to develop a Core-to-F translator for the purpose of testing the prototype further.[1] I plan to implement the full type-inference-based deforestation algorithm, incorporating this prototype, and study its performance in relation to other deforestation algorithms, as well as the ways in which deforestation interacts with other compiler optimizations. My work will contribute to a better understanding of how and when deforestation should be applied to programs.

References

[1] Chitil, Olaf, "Type-Inference Based Deforestation of Functional Programs", unpublished Ph.D thesis, 2000.

[2] Dasgupta, Sanjoy, "Learning mixtures of Gaussians", IEEE Symposium on Foundations of Computer Science, 1999.

[3] Andrew Gill, John Launchbury, and Simon Peyton Jones, "A Short Cut to Deforestation", Proceedings of the Conference on Functional Programming Languages and Computer Architecture (FPCA '93), pp. 223-232, 1993.

[4] Launchbury, John, and Tim Sheard, "Warm Fusion: Deriving Build-Catas from Recursive Definitions", Proceedings of the Conference on Functional Programming Languages and Computer Architecture (FPCA '95), pp. 314-322, 1995.

[5] Wadler, Philip, "Deforestation: transforming programs to eliminate trees", *Theoretical Computer Science*, v. 73, pp. 231-248, 1990.

[1] The prototype handles programs in F, a modified version of Core, the intermediate language used by the Glasgow Haskell Compiler; a method of translating Haskell programs into F, via Core, is necessary to substantially test the prototype