A Study of Spilling and Coalescing in Register Allocation as Two Separate Phases

Florent Bouchez
École Normal Supérieure de Lyon

Thesis reviewed by Keith D. Cooper, Rice University

Introduction

Florent Bouchez's thesis presents an in-depth examination of the problems that arise in building a sound and effective register allocator. His work is both timely and thorough. His work expands our understanding of the complexity of the underlying problems. It clarifies misconceptions that are widely held in the compiler-construction community. It uses the theoretical results to proposed strong and practical heuristic techniques for register allocation that improve on current practice. Finally, he applies the same insights to translation out of SSA form and brings the same rigor and clarity to that problem that he brought to register allocation. Bouchez's thesis is an important piece of work that deserves widespread attention. It will change the way that we understand these two problems and the way that we try to solve them.

For several decades, the compiler community has focused on graph coloring as a paradigm for register allocation. This notion goes back, in the Soviet literature, to Lavrov in the 1950s and Ershov in the 1960s. In the west, the ideas were popularized by John Cocke and, subsequently, by Greg Chaitin and his colleagues at IBM who built the first practical graph-coloring register allocator. While graph coloring did provide a significant improvement over the ad-hoc techniques that preceded it, coloring did not solve all of the problems necessary to ensure effective register use in real programs. In particular, Chaitin's scheme and its descendants (including our work) suffer from lumping together all of the aspects of register allocation, including spill choice, spill placement, live-range splitting, rematerialization, copy coalescing, and register assignment, into a single over-arching paradigm.

In recent years, a series of theoretical results have provided glimpses of a deeper structure to the complexity of the problems that underlie register allocation and have provided hints that a more partitioned and nuanced approach to the problem might yield both deeper insight and better practical algorithms. Bouchez's thesis is the most complete exploration of that theory to date. He provides a tour of the problems that arise in spilling values and in coalescing move operations; he firmly establishes the complexity of these various problems; and he uses the results to drive his invention of strong, practical algorithms.

Bouchez's thesis is a significant achievement in the underlying theory of register allocation and in the application of that theory to practical problems, both in allocation itself and, more generally, in code optimization. He has added significantly to our understanding and to the store of techniques that we have for attacking this complex and important problem.

Summary of Major Results

Bouchez begins, in Chapter 2, by laying out the basic terminology of program optimization and of register allocation. He provides detailed and specific definitions for the various aspects of register allocation. Of particular interest are Definitions 2.8 and 2.11 that specify the meaning of interference, and the critical subclasses of interference graphs that form the backbone of the proofs in the thesis—specifically interval graphs, chordal graphs, greedy-*k*-colorable graphs, and general graphs. Finally, he defines the basic schemes of both Chaitin's approach and the George and Appel Iterated Register Coalescing approach.

Chaitin's work and its proof of NP-completeness form the critical background for this work. Chaitin et al. proved that the overall allocation problem is reducible to graph coloring (a classic NP-complete problem) and that, for any graph, we can construct a program with the corresponding interference graph. Unfortunately, this proof was interpreted in many minds as equating all of register allocation, which has many subtle sub-problems, as equivalent to graph coloring. In 2005, several groups discovered that the SSA form of the program yields chordal graphs. Since chordal graphs can be colored in polynomial time, this discovery started a new round of exploration in both the theory and practice of register allocation. Bouchez's thesis makes critical contributions to both the theory and the practice of this new school of register allocation.

Bouchez devotes Chapter 3 of his thesis to exploring the boundaries and limitations of Chaitin's original NP-completeness proof. In particular, he explores the impact of live-range splitting (or live-range choice) on the complexity of the problem. He identifies several instances in which the problem falls into the realm of polynomial time algorithms. His in-depth analysis suggests that three critical issues make register allocation NP-complete.

- The presence of unsplit critical edges in the control-flow graph creates multiplexing regions where values are hard to color.
- The optimization of spill costs introduces the problem of selecting one or more live ranges to spill; that problem is, in general, NP-complete.
- Optimal coalescing is, in general, NP-complete.

While these results might be viewed as pessimistic—allocation is essentially the product of multiple different NP-complete problems—they also suggest a new decomposition for a register allocator. Bouchez's decomposition is as follows:

1. Convert the code into SSA form
2. Spill values until register pressure is uniformly less than *R*
3. Coalesce and color to obtain an assignment of colors to values
4. Translate out of the colored SSA form while inserting as few additional copies as possible.

The first step is well covered in the literature. The remaining chapters form the subject of Chapters 4 through 7 of this thesis.

In Chapter 4, Bouchez presents a theoretical study on complexity of spilling in a graph-coloring register allocator. Not surprisingly, the results show that spilling is hard. In particular, while the spill-everywhere can be solved optimally on an interval graph, it is NP-complete on chordal graphs and more general graphs. The incremental versions of spill-everywhere have the same complexities. Adding holes to the live ranges to account for the small live ranges left by spilling on a RISC-like architecture makes the problem harder. Thus, the main result of this chapter is pessimistic; while SSA simplifies coloring, it does not simplify spilling.

In Chapter 5, Bouchez examines the complexity of coalescing register move operations. Coalescing is more important today than it was in Chaitin's day, in large part because of the large cost differential between a move operation and a load or a store operation. Since coalescing two live ranges can reduce the degree of all their common neighbors, good coalescing decisions can reduce the number of spills. At the same time, new problems have made strong coalescing techniques important subjects of practical interest; for example, out-of-SSA translation introduces myriad copies that include some hard-to-coalesce cases.

Bouchez examines four versions of coalescing from the literature: aggressive coalescing, conservative coalescing, the incremental version of conservative coalescing, and optimistic coalescing. Aggressive, conservative, and optimistic are all shown to be NP-complete in their reasonable forms. However, incremental conservative coalescing can be solved on a chordal graph using a technique that recreates the chordal property after the coalesce; this proof directly suggests the chordal coalescing heuristic used in Chapter 6.

In Chapter 6, Bouchez uses insights from the complexity results to build new strong heuristics for conservative coalescing. These include brute-force conservative coalescing, a chordal-graph-based incremental conservative coalescing, and a novel optimistic coalescing scheme. Finally, careful examination of sub graphs in which the compiler can make an optimal decision led him also to a strong aggressive coalescing heuristic.

He compares his new techniques against classic heuristics from the literature. Using the graphs from the George-Appel Coalescing Challenge, he shows that brute-force conservative coalescing, with the chordal improvement, is more effective than the other known methods. Further, he shows a strong impact from tie breaking—the order in which affinities are considered for coalescing.

(Note that the winning heuristic, brute-force with chordal improvement, derives directly from two of the earlier complexity results: the identification of a subclass of graphs that are greedy-k-colorable in polynomial time and the proof of the polynomial time complexity for incremental coalescing on a chordal graph.)

Chapter 7 addresses one of the significant problems introduced by the use of SSA-based techniques for register allocation: the issue of out-of-SSA translation for already allocated code. The compiler faces three challenges in this problem: dealing with the problems of out-of-SSA translation, handling the placement of copies on critical edges, and inserting register-permutation operations to regions in the code where no spare register is available. Bouchez shows that a uniform approach to handling all three has the potential to solve them simultaneously. He models the inserted copy operations as parallel copies. He develops a novel scheme for decomposing a general parallel copy into a copy that deals with duplication and a strictly reversible permutation. He then explores the potential for moving permutations across edges and into the interiors of blocks with his region-recoloring technique. He introduces a general notion of compensation code based on the reversibility of the permutations. This unique decomposition, coupled with permutation motion, have the potential to significantly change the cost and placement of code to implement parallel copies.

Taken together, the chapters of Florent Bouchez's Ph.D. thesis form the most complete exploration of the theory of register allocation that I have seen. The theoretical results suggest novel directions for the structure of register allocators and for the solution of several of the critical sub problems that arise in building a register allocator. This work adds significantly to our understanding of the problem and to our store of techniques for attacking its many complex parts.

Brief Biographical Sketch of the Reviewer

Dr. Keith Cooper is the L. John and Ann H. Doerr Professor of Computational Engineering at Rice University, with appointments in the Department of Computer Science and the Department of Electrical and Computer Engineering. He chaired the Computer Science Department from July 2002 through June 2008.

Dr. Cooper's research has focused on the analysis and transformation of programs. Two areas of work are of relevance to this thesis.

- Cooper worked (with Preston Briggs, Tim Harvey, Ken Kennedy, and Linda Torczon) on a series of modifications to Chaitin's graph-coloring register allocator that are widely used today. With his students, he has studied other variants of register allocators, including Chow's allocator (with Dave Peixotto), the Koblenz-Callahan allocator (with Jason Eckhardt), and adaptive versions of the Chaitin-Briggs scheme (with Anshuman DasGupta and with Donghua Liu).
- Cooper did early work on out-of-SSA translation. Cliff Click, Cooper's second Ph.D. student, was one of the first people to identify the problem and isolate it into small examples. With his students Preston Briggs, Taylor Simpson, and Timothy Harvey, Cooper published one of the first papers to identify the difficulties in out-of-SSA translation and to pose a set of solutions to them.

These two problems play a significant role in Mr. Bouchez's thesis. He poses, for each of them, novel and elegant insights and solutions.