# Partial Redundancy Elimination using Lazy Code Motion

Sandeep Dasgupta[1] Tanmay Gangwani[2]

October 28, 2015

[1]Electronic address: sdasgup3@illinois.edu
[2]Electronic address: gangwan2@illinois.edu

## Value-Number driven code motion

We implement an iterative bit vector data flow algorithm for PRE. We initially implemented the flow equations from the Lazy Code Motion paper. This set included a total of 13 bit vectors for each basic block - 2 for block local properties ANTLOC and TRANSP, and 11 for global properties. These equations, however, could only be applied to single instruction basic blocks. We therefore, derived a new set of equations which are motived by later work[**?**] of the same authors. This set of equations apply to maximal basic blocks and entails a total of 19 bit vectors for each basic block in our current implementation - 3 for block local properties ANTLOC, TRANSP, XCOMP and 16 for global properties. We include the equations in appendix A and B. In appendix C, we provide our generalized data flow framework, and show how each PRE equation maps to the framework. We call the algorithm value-number driven because each slot in each of the bit vectors is reserved for a particular value number rather than a particular expression. Also, we make the observation that a large number of expressions in the program only occur once, and are not useful for PRE. Hence to further optimize for space and time, we only give bit vector slots to value numbers which have more than one expression linked to them.

### Local CSE

For our data flow equations to work correctly, a local CSE pass has to be run on each basic block. Basically, this path removes the redundancies inside straight line basic block code and sanitizes it for the iterative bit vector algorithm. This idea is borrowed from [**?**]. We perform this step before calling our data flow framework.

# Bibliography

[1] K. heinz Drechsler and M. P. Stadel, *A variation of knoop, rothing, and steffen's lazy code motion.*

[2] J. Knoop, O. Rüthing, and B. Steffen, *Lazy code motion*, in Proceedings of the ACM SIGPLAN 1992 Conference on Programming Language Design and Implementation, PLDI '92, New York, NY, USA, 1992, ACM, pp. 224–234.