Abstract: Over the past several decades of compiler research, there have been great successes in automatically optimizing regular programs, which operate over dense matrices and arrays. Tackling irregular programs, which operate over pointer-based data structures such as trees and graphs, has been much harder, and has mostly been left to ad hoc, application specific methods. In this talk, I will describe efforts by my group to automatically optimize a particular class of irregular applications, those that traverse trees. The key insight behind our approach is an abstraction of data structure traversals as operations on vectors. This abstraction lets us design transformations, predict their behavior and determine their correctness. I will present several transformations that my group has developed that enhance locality and enable vectorization, as well as analyses that prove the transformations’ soundness. I will conclude by discussing some future directions that we are exploring.

Biography: Milind Kulkarni is an associate professor in the School of Electrical and Computer Engineering at Purdue University. His research focuses on developing languages, compilers and systems that can efficiently and effectively exploit locality and parallelism in complex applications on complex computation platforms. Before joining Purdue, he was a postdoctoral research associate at the University of Texas at Austin from May 2008 to August 2009. He received his Ph.D. in Computer Science from Cornell University in 2008, where he was a Department of Energy High Performance Computer Science (HPCS) Fellow. Prior to that, he received his M.S. in Computer Science from Cornell University in 2005, and BS degrees in Computer Science and Computer Engineering from North Carolina State University in 2002. He received the NSF CAREER award in 2012 and the Department of Energy Early Career Research Award in 2013 for his work on optimizing irregular applications. He was awarded the Presidential Early Career Award for Scientists and Engineers in 2016. He is a member of the ACM and the IEEE Computer Society.