Software synthesis is a technique for automatically generating code from a given specification. The goal of software synthesis is to **make software development easier** while increasing both the **productivity of the programmer** and the **correctness of the produced code**. In this talk I will present an approach to synthesis that relies on the use of automated reasoning and decision procedures. In addition, I will describe a procedure that **synthesizes code snippets** from specifications implicitly given in the form of type constraints. As there can be multiple solutions, more than one code snippet can be a good candidate. We use an additional weight function to **rank the derived snippets** and direct the search for them. In practical evaluation, our technique scales to programs with thousands of visible declarations in scope and succeeds in 0.5 seconds to suggest program expressions of interest. I will conclude with an outlook on possible future research directions and applications of synthesis procedures.

**Ruzica Piskac** is an assistant professor (tenure-track) at Yale, Computer Science Department. Her research interests span the areas of programming languages, software verification, automated reasoning, and code synthesis. A common thread in Ruzica’s research is improving software reliability and trustworthiness using formal techniques. Prior to joining Yale, Ruzica was an independent research group leader at the Max Planck Institute for Software Systems (MPI-SWS) in Germany (January 2012-August 2013). She was head of the Synthesis, Analysis and Automated Reasoning Group. Ruzica did her graduate studies at EPFL, Switzerland under the supervision of Viktor Kuncak.