

# Transforming Undecidable Synthesis Problems into Decidable Problems

Bernd Finkbeiner

Universität des Saarlandes  
finkbeiner@cs.uni-saarland.de

Synthesis holds the promise to revolutionize the development of complex systems by automating the translation from specifications to implementations. Synthesis algorithms are based on the same level of mathematical rigor as verification algorithms but can be applied at earlier development stages, when only parts of the design are available. Given a formal specification of the desired system properties, for example in a temporal logic, we determine if the partial design can be completed into a full design that satisfies the properties.

For general distributed systems, the synthesis problem is undecidable. However, there has been a sequence of discoveries where the decidability was established for specific system architectures, such as pipelines and rings, or other restrictions on the problem, such as local specifications. Encouraged by these findings, new specification languages like Coordination Logic aim for a uniform treatment of the synthesis problem.

In this talk, I will review several techniques that transform undecidable synthesis problems into decidable problems. Compositional synthesis uses a proof rule to reduce an undecidable synthesis problem into several decidable synthesis problems. Bounded synthesis transforms the synthesis problem into a decidable search problem by introducing a bound on the size of the implementation. Lazy synthesis transforms the synthesis problem into a sequence of constraint solving problems, each decidable but increasingly complex, until an implementation is found.