## Towards Concolic Testing for Curry

## Jan Rasmus Tikovsky

## Institut für Informatik, CAU Kiel, D-24098 Kiel, Germany, jrt@informatik.uni-kiel.de

**Abstract** Testing is the most commonly used method towards ensuring the correctness of programs. But writing test cases covering all possible execution paths manually can be a rather tedious task especially as programs become more complex.

With the use of automated testing tools this task can be considerably facilitated. In the last years, concolic testing tools like jCUTE for Java or CutEr for Erlang have gained more and more popularity. Concolic testing is a combination of concrete and symbolic program execution: During the concrete execution of a program with some random input symbolic constraints are collected on every branch decision. Then, these constraints are systematically negated and solved applying a constraint solver. In the next step, the computed solutions are mapped to new input data driving the concrete execution along yet unvisited execution paths. An appropriate search strategy is used to select the path constraints to be negated. This process is repeated until all execution paths are visited.

In this work, we propose an approach towards the concolic testing of functional logic programs considering the functional logic language Curry. We present a concolic interpretation of Curry programs gathering symbolic constraints on the constructor decisions which are made during the evaluation of case expressions. We plan to apply an SMT solver to solve these path constraints with the aim of computing new input data for our program driving the execution along yet unvisited program paths.