

Statechart Analysis with Symbolic PathFinder

Corina S. Păsăreanu
CMU-SV/NASA Ames
NASA Ames Research Park, MS 269-2, PO Box 1
Moffett Field, CA 94035
Email: corina.s.pasareanu@nasa.gov

II. ANALYSIS WITH SYMBOLIC PATHFINDER

Symbolic PathFinder (SPF) [2] is an analysis tool for Java bytecode that performs symbolic execution to generate test cases that achieve high test coverage. Symbolic execution is a systematic program analysis that uses symbolic values instead of actual data inputs and symbolic expressions to represent the values of program variables. The state of a symbolically executed program includes the symbolic values of program variables, a path condition (PC), and a program counter. The path condition is a Boolean formula over the symbolic inputs, encoding the constraints which the inputs must satisfy in order for an execution to follow the particular associated path. These conditions are solved using off-the-shelf constraint solvers to generate test cases guaranteed to exercise the analyzed code. Symbolic PathFinder generates both test vectors and test sequences; the latter are necessary for testing looping, reactive programs, such as the ones translated from the Statechart models. During test case generation, SPF checks the properties of the code, expressed as assertions or property automata.

We have applied SPF to the analysis and test case generation of Statecharts in Polyglot. The analysis uncovered subtle interaction errors between components modeled with Statecharts, for the flight software developed for NASA Exploration. To increase the speed of our analysis, we are investigating program specialization via symbolic execution. This involves using SPF to specialize the Polyglot semantic modules with respect to particular Statechart models. Our preliminary results are encouraging, showing 3x improvement in analysis time, with 10x fewer instructions being executed by SPF.

ACKNOWLEDGMENT

This is joint work with D. Balasubramanian, M. Whalen, G. Karsai, T. Pressburger and M. Lowry.

REFERENCES
