A symbolic evaluator for JavaScript

Mehdi Bouaziz
École normale supérieure, Paris

Joint work with Andrey Chudnov and David Naumann
Stevens Institute of Technology

October 2010 — February 2011

Motivation

Static analysis of web programs written in JavaScript:

- information-flow security analysis
- testing, debugging
- program proving
- optimization

Symbolic evaluation

- Like evaluation but values can be symbolic expressions
- Keep track of the conditions of the execution paths
- The result is a list of tuples:
  (*path condition*, *symbolic result*)
- Uses an SMT solver to eliminate unfeasible paths

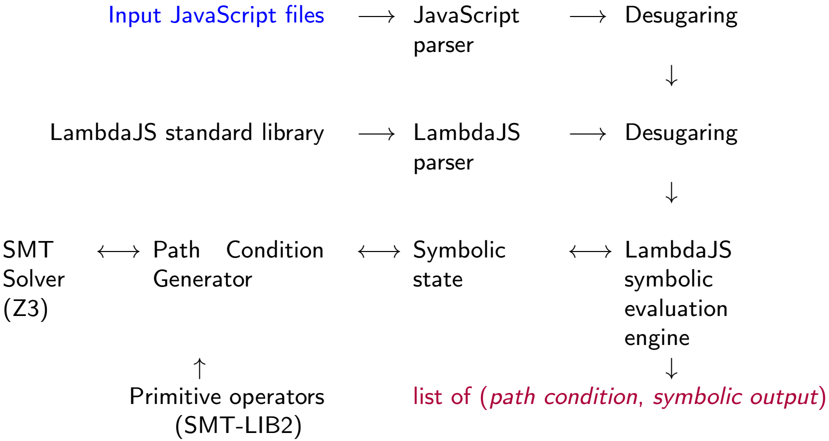
Limitations:

- Explosion of the number of paths
- Non-termination

JavaScript: a highly dynamic language

- No typing, implicit casts
- *eval* function
- Object property names can be dynamically computed
- Most data are strings
- Scripts are embedded in HTML pages
- . . .

Architecture



Input JavaScript files → JavaScript parser → Desugaring

↓

LambdaJS standard library → LambdaJS parser → Desugaring

↓

SMT Solver (Z3) ⟷ Path Condition Generator ⟷ Symbolic state ⟷ LambdaJS symbolic evaluation engine

↑ Primitive operators (SMT-LIB2)

↓ list of (*path condition*, *symbolic output*)

Demo: a currency converter

Demo!

Your opinion: a name for it

- Jsx
- Moreas
- Syjex
- Jaxemys
- your choice

Thank you!

How to find it:

- My e-mail address: mehdi.bouaziz@ens.fr
- Git repository: http://mehdi.bouaziz.org/git/jsx/
- Web site (soon?): http://mehdi.bouaziz.org/jsx/