[Demo]

MoCHi: Software Model Checker for a Higher-Order Functional Language

Ryosuke Sato  Hiroshi Unno  Naoki Kobayashi
Tohoku University  University of Tsukuba  University of Tokyo
MoCHi: Software **Model Checker for Higher-Order Functional Programs**

- Based on *higher-order model checking* [Ong 2006, Kobayashi 2009], *predicate abstraction*, and *CEGAR* [PLDI 2011]
- Full automatic

OCaml program ➔ assertions never fail ➔ Refinement intersection types as a certificate ➔ can fail ➔ Counterexample
Supported Language Features

• Booleans, integers, tuples
• Recursion
• Higher-order functions
• Exceptions (modulo some restrictions)
• Algebraic data types (includes user-defined data types)
Supported Properties

• The assertions in the program never fail
• The pattern matches in the program are exhaustive
• An uncaught exception does not occur
Internals of MoCHi

OCaml program

Encoding algebraic data types and exceptions as functions

Program with integers, recursion, and higher-order functions

Verification for PCF-like language [PLDI 2011]
Internals of MoCHi

OCaml program

Encoding algebraic data types and exceptions as functions

Program with integers, recursion, and higher-order functions

Predicate abstraction

Boolean program

HO Model checking

Counterexample

Unsafe

Feasibility checking

Predicate Discovery

Safe

[PLDI 2011]
## Comparison with Related work

<table>
<thead>
<tr>
<th></th>
<th>Annotations</th>
<th>Counterexample discovery</th>
<th>Intersection types (Context-sensitivity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoCHi</td>
<td>Nothing</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Liquid types</td>
<td>Predicates used in refinements</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>[Rondon et al.]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DML</td>
<td>Refinement types of recursive functions</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>[Xi et al.]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMC</td>
<td>Nothing</td>
<td>✓?</td>
<td>✗</td>
</tr>
<tr>
<td>[Jhala et al.]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusion & Future work

• We have implemented MoCHi, a verifier for a subset of OCaml with
  • base types, tuples, higher-order functions, recursions
  • exceptions, algebraic data types

• Future work includes:
  • Supporting a larger subset of OCaml
  • To be more scalable