Deriving a Symbolic Executor for Definitional Interpreters Suitable for the Study of Heuristics

Laura Pircalaboiu, Casper Bach Poulsen & Cas van der Rest

Research Project CSE3000

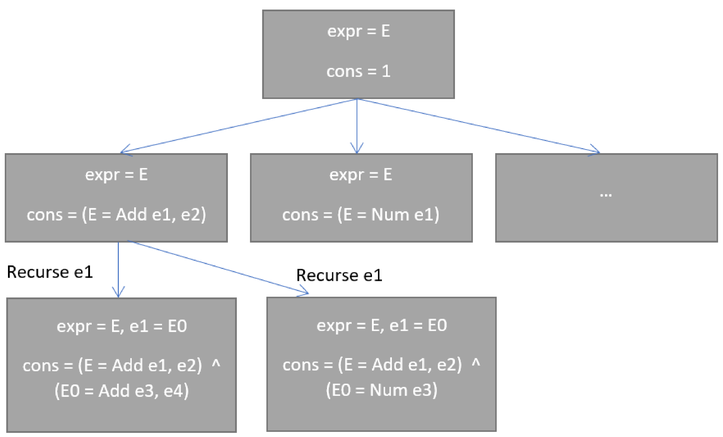
Introduction

- Student submissions in courses are hard to manually evaluate
- Unit testing insufficient
- Mensing approach to symbolic execution– effective, but we want to extend it
- Use as starting point: interpreters like defined in PLAI [2].

The goal of this research project is to determine whether any two given definitional interpreters are equal or not using symbolic execution.

We want to create a simple and extensible approach to symbolic execution.

Intuition: Building Execution Trees



Method: Intermediate Representation

How do we encode interpreters to ensure consistency in our results and extensibility of the approach?

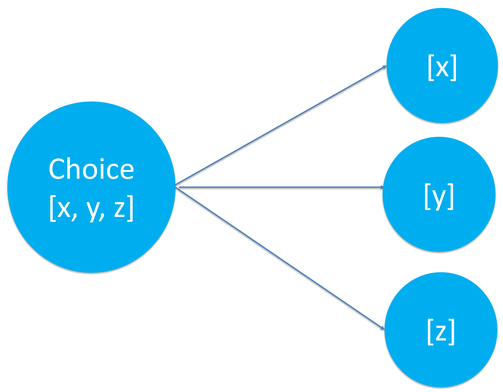

```
data Expr = Num Int |
   Add Expr Expr

eval :: Expr -> Int
eval (Num i) = i
eval (Add e1 e2) =
   eval e1 + eval e2
```

```
eval =
   ([e = Num(i)].
         return t)
   + ([e != Num(i),
      e = Add(e1, e2)]
   .recurse e1 as i1
   .recurse e2 as i2
   .return +(i1, i2))
```

Method: Small Step Transition Function

Idea adopted from Mensing et al. [1]:



Results

Actual \ Predicted	True	False
True	4	0
False	10	20

We compared 8 interpreters that belong to 3 equivalence classes, for a total of 34 test cases.
The results are reported in the confusion matrix above.

Discussion & Future Work

The approach works for finding trivial bugs, such as wrong order of variables or typos, but gives false negatives in the case of equivalent interpreters that have a different branch order.

Possible future improvements are:

- Extensions to the programming languages
- Usage of Heuristics and/or branch pruning
- Ability to run two interpreters in (real) lock-step

References

1. A. D. Mensing, "From Definitional Interpreter to Symbolic Executor ,"en, p. 10, 2019.
2. S. Krishnamurthi, "Programming Languages: Application and Interpretation," en, p. 207.

TU Delft
Delft University of Technology

l.a.pircalaboiu@student.tudelft.nl