

Interactive Stepper for Expression with Holes

YANJUN CHEN*, University of Michigan, USA

1 INTRODUCTION

Stepper that can display all immediate states of a program is a useful tool for debugging and education. It simplifies expression step by step so that people can track the evaluation procedure easily. Hazel, a programming language environment developed by [citation Cyrus Omar], allows to evaluate incomplete programs. It defines a special variable called hole indicating the place that needs to fill. In this paper, we want to develop an interactive stepper for Hazel. Take the following programs as examples:

```
fibo 1
```

```
TYPE OF RESULT: Int
```

```
case 1:1
```

```
| 5 ⇒ ...
```

```
| 6 ⇒ ...
```

```
end
```

(a) Example of paused environment

```
fibo 7 + (6 * 9) + λx.{x + 1} 44
```

```
TYPE OF RESULT: Int
```

```
<fn> 7 + 6 * 9 + <fn> 44
```

(b) Example of multiple environment

Fig. 1. Two programs in Hazel

Figure 1a is an example of expression with a hole. The expression is already simple enough and it is unnecessary to expand cases since the parameter has not provided especially when there are many cases in the function. Therefore, stepper needs to detect this situation so that it will pause until user requires it to progress.

Figure 1b shows an expression with three parts. The left part is heavy since it takes many steps to evaluate while user might just want to debug the right lambda part. According to the regular dynamics, there should contain only one way to step for any non-final programs. So, user has to click many times to reduce the left one in order to debug the right one. In the Figure 1b, the green boxes contain subexpressions that can be evaluated. So, our solution is just allowing the multiple evaluation contexts so that user can choose the subexpression they need.

The interactive stepper develops pausing environment and a simple algorithm to decompose multiple evaluation contexts. Furthermore, it also has a webview so that user can click green or yellow boxes to step expressions.

Contributions. The contributions of this paper are: (1) a pausing environment in section ??; (2) an algorithm to decompose multiple evaluation contexts in section ??.

*Research advisor: Cyrus Omar; ACM student member number: ; Category: undergraduate

Author's address: Yanjun Chen, University of Michigan, USA, yanjunc@umich.edu.