

エラー生成クイズの提案と実施様子の観察の紹介

角田 和広 増原 英彦 叢 悠悠 (東京工業大学)

背景・動機

- ・ 初学者はエラーメッセージをよく読まない/読めないのではないかな?

[例]

```
1 case class Color(m:Int,y:Int,c:Int)
2
3 def downsat(a:Color): Color = {
4   a match {
5     case Color(b,c,d) => Color(c-10,d-10)
6   }
7 }
```

理想: この部分のcが不足していると理解して、**ここ**を直す

現実: この部分だけ見て、**この辺**を適当に変更する

missing argument for parameter c of method apply in object Color: (m:Int, y:Int, c:Int): Playground.Color

→ 初学者にエラーメッセージをしっかり読んでほしい

予備実験

[目的] エラー生成クイズを実施する際に大きな欠陥がないかどうかの確認
+ 実施時に解答者がエラーメッセージを読むかの確認

東工大の学生に3名に予備実験(エラー生成クイズの実施観察 + インタビュー)を実施

- ・ エラー生成クイズの実施する際の大きな欠陥なし
- ・ エラー生成クイズの解答時に、確かにエラーメッセージを読むことを確認
- ・ 「自発的な単語の検索」、「言語の興味促進」等の興味深い特徴を確認

提案: エラー生成クイズ

エラーメッセージを必ず読む問題形式として**エラー生成クイズ**を提案

[概要] **正しいコード**の赤枠1ヶ所を変更し、**エラーメッセージ**¹を発生させる形式

[例] 以下のコードの赤枠1ヶ所だけ変更して、

```
1 def and(a:Boolean,b:Boolean): Boolean = {
2   a && b
3 }
```

以下のエラーを発生させる。

✖ value && is not a member of Int

元コード

(この正しいコードを変更する)

目的エラーメッセージ

(これを発生させることを目指す)

以下のコードの赤枠1ヶ所だけ変更して、

```
1 def head(l1:List[Int]): Int = {
2   l1 match {
3     case Nil => 0
4     case x::xs => x
5   }
6 }
```

以下の警告を発生させる。

⚠ Unreachable case

1*: 警告メッセージも同様に扱う
(目的警告メッセージ)

展望と課題

- ・ エラー生成クイズ作成の自動化
 - ミュータントテストで用いられるミュータント生成の応用^[亀井 '21]や compiler fuzzing で用いられる simplification^[石浦 '16]技術
- ・ エラー生成クイズの効果測定
 - エラーメッセージの理解度の変化などについてデバッグ型演習と比較実験
- ・ エラー生成クイズ向きでないエラーメッセージの活用への工夫方法