# 異なるFault Localization手法の欠陥種別に基づく比較評価

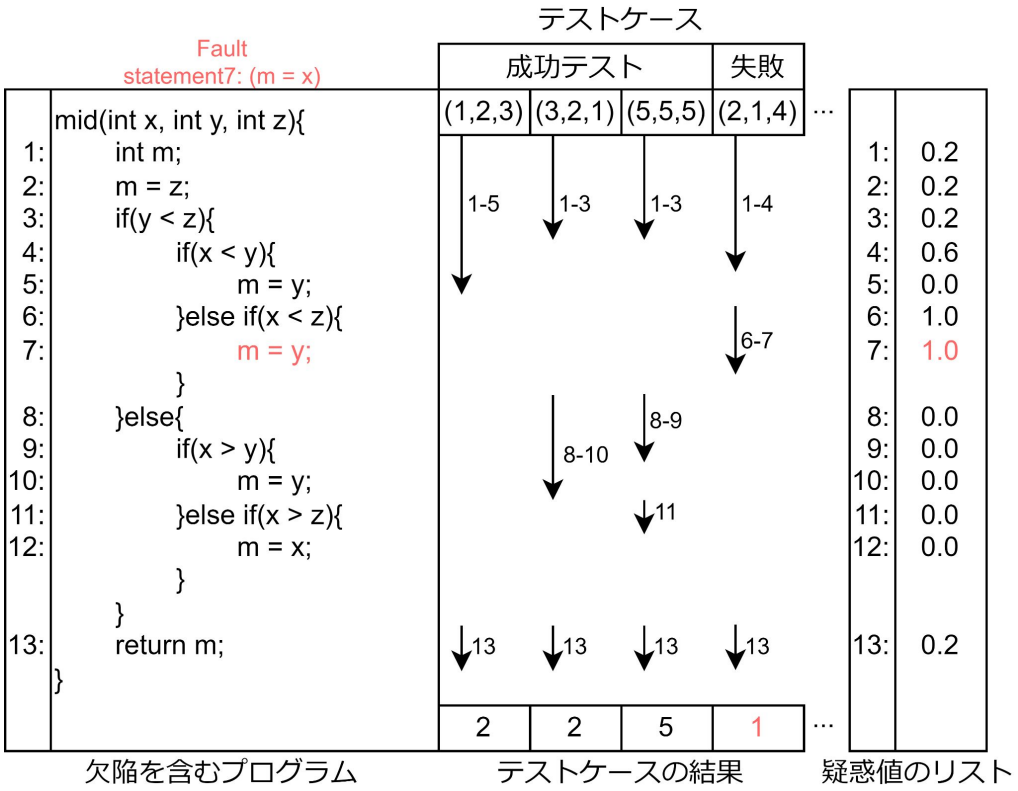
鷲崎研究室 M0
高井悠宇
1W182187-0

# 背景

Fault Localization (FL)

- プログラム内の欠陥を自動的に推定する技術
- アルゴリズムの違いから様々な手法が提案されている
  - e.g. Spectrum Based Fault Localization(SBFL), Mutation Based Fault Localization(MBFL)
- 欠陥を含むプログラムとテストスイートが入力
- statement毎の怪しさの度合い(疑惑値)を出力

# 背景: Spectrum Based Fault Localization(SBFL)



# 提案と研究課題

欠陥種別に応じて検出性能は変化しないか？

- アルゴリズムとの親和性
- 単純、複雑な欠陥

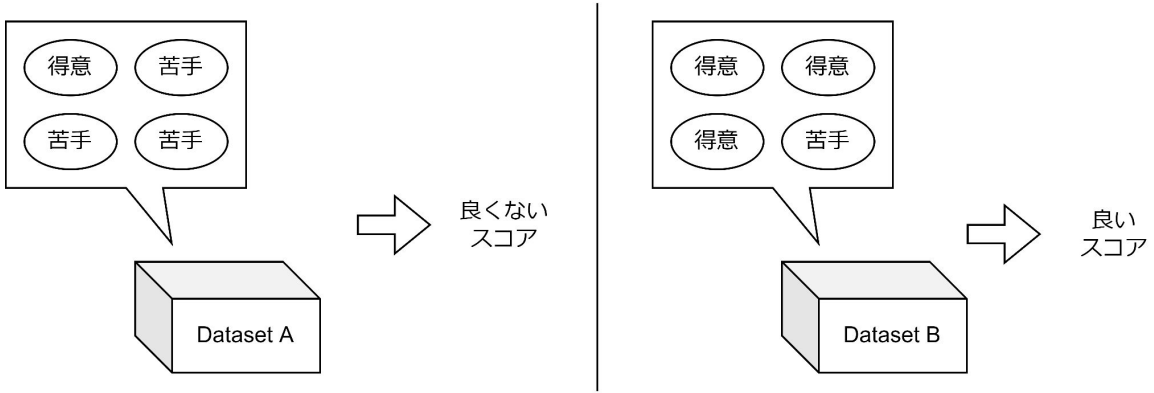
現状の評価

- 利用する欠陥は適当に設定される
- 欠陥種別という観点からの評価は行なわれない

# 提案と研究課題

## 現状の評価における問題点

- データセットによって検出性能が変化する可能性(RQ1, RQ2)
- 正しくFL手法の性格・特徴が理解できない
  - 偶然得意(不得意)な欠陥種別ばかりで構成されている可能性
- 未知のソフトウェアに対して実用的でない(汎用性がない)
  - 苦手な欠陥種別ばかりが含まれていたらどうするのか



# 提案と研究課題

どの手法でも検出しにくい欠陥は存在するか？(RQ3)

- 様々なFL手法を組み合わせた複合的手法の提案 [1][2]
  - どの手法でも検出しにくい欠陥は複合的手法でも検出しにくい

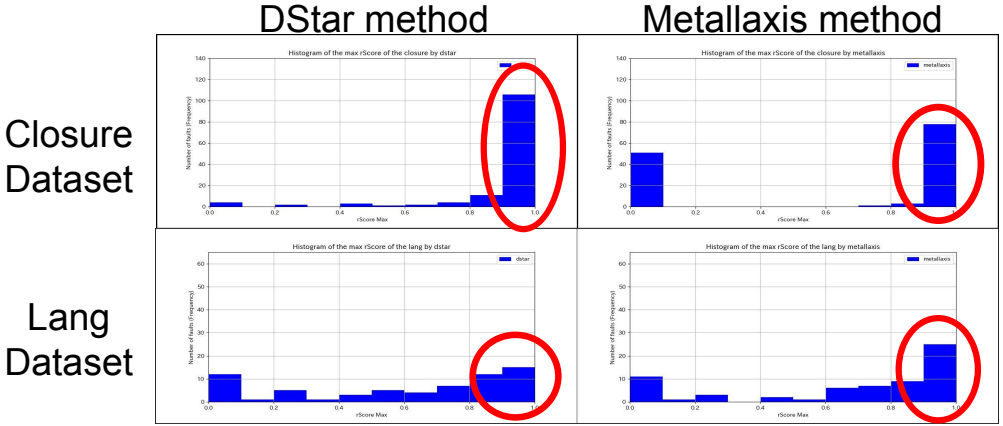
欠陥種別を分類した際に何か顕著な特徴が見られるか？(RQ4)

  - ODC分析を用いて手動で分類

# 実験結果(RQ2)

RQ2. データセットによって大きく性能が変化するFL 手法は存在したか？

- 標準化順位(rScore)が0.9より大きい = 正確に欠陥箇所を推定できた
- 複数のFL手法について検出性能の変化がみられた




rScore が 0.9より大きい欠陥の割合

| | DStar | Metallaxis |
|---|---|---|
| Closure | 80% | 59% |
| Lang | 22% | 37% |

# 実験結果(RQ3, RQ4)

RQ3. 手法によらず検出しにくい欠陥は存在するか？

- 全体の18%の欠陥については手法によらず標準化順位が0.9以下であった

RQ4. 特定の欠陥種別において検出しにくい(しやすい)手法は存在するか？

| Defect Type | ochiai | DStar | Metallaxis | MUSE | slicing | stack trace | predicate switching |
|---|---|---|---|---|---|---|---|
| all | 38% | 38% | 48% | 24% | 23% | 18% | 6% |
| assignment/initialization | 47% | 47% | 63% | 16% | 26% | 26% | 11% |
| checking | 31% | 31% | 41% | 26% | 31% | 15% | 10% |
| algorithm/method | 45% | 45% | 55% | 30% | 20% | 25% | 0% |
| timing/serialization | 60% | 60% | 60% | 20% | 0% | 0% | 0% |
| interface/message | - | - | - | - | - | - | - |
| relationship | - | - | - | - | - | - | - |
| function/class/method | 25% | 25% | 33% | 25% | 8% | 8% | 0% |
| GUI | - | - | - | - | - | - | - |

# まとめと今後の展望

1. FL手法の性能がデータセットにより異なるという結果が得られた
2. 欠陥種別という観点からFL 手法の評価を行った

展望

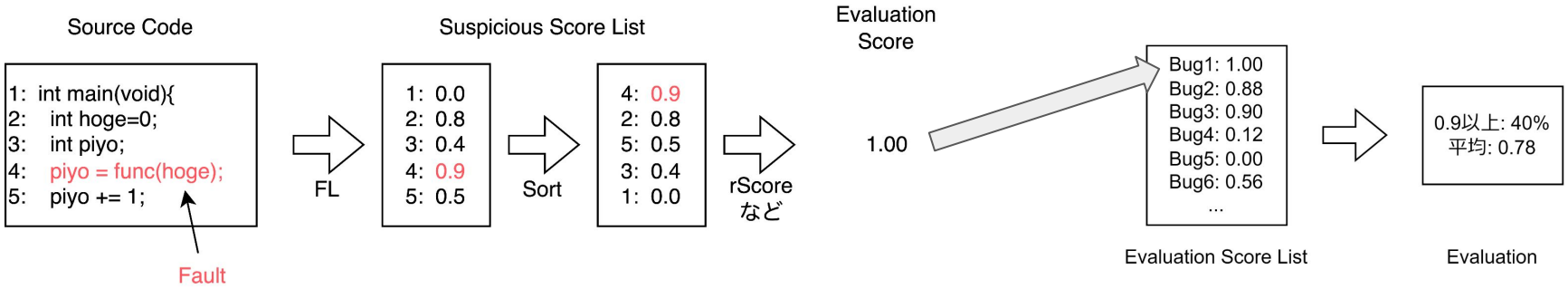
- RQ3, RQ4 の更なる考察
- ODC分析という分類方法についての再検討
  - 欠陥種別の改善
  - 自動化
- 特に複合的なFL手法など、さらなるFL手法の追加実験

# 参考文献

[1] 鷲崎弘宜, “機械学習を中心としたai活用によるソフトウェアの品質保証,” inシステム/制御/情報, vol.66, no. 5, 2022, pp. 1–7.
[2] D. Zou, J. Liang, Y. Xiong, M. D. Ernst, and L. Zhang, “An empirical study of fault localization families and their combinations,” IEEE Transactions on Software Engineering, vol. 47, no. 2, pp. 332–347, 2021.

# 付録

## FL手法の評価


Source Code
1: int main(void){
2: int hoge=0;
3: int piyo;
4: piyo = func(hoge);
5: piyo += 1;
Fault
FL
Suspicious Score List
1: 0.0
2: 0.8
3: 0.4
4: 0.9
5: 0.5
Sort
4: 0.9
2: 0.8
5: 0.5
3: 0.4
1: 0.0
rScore
など
Evaluation
Score
1.00
Bug1: 1.00
Bug2: 0.88
Bug3: 0.90
Bug4: 0.12
Bug5: 0.00
Bug6: 0.56
...
Evaluation Score List
0.9以上: 40%
平均: 0.78
Evaluation

# 付録

## ODC分析

- 直交した属性を用いてタグ付けを行い分類する方法
  - Trigger
    - いつ欠陥を発見したか
    - e.g. Variation, Interaction
  - Defect Type
    - 修正した欠陥の種類
    - e.g. Checking, Algorithm/Method
  - Qualifier
    - 欠陥埋め込みの種類
    - e.g. incorrect, missing
  - Age
    - いつ作りこまれたか
    - e.g. new, base

# 付録

表 7　少なくとも 1 つ以上の手法で標準化順位が $n$ 以上であった欠陥数と割合

| $n$ | 欠陥数 | 割合 | 合計 |
|---|---|---|---|
| 0.9 | 292 | 82% | 357 |
| 0.95 | 259 | 73% | 357 |
| 0.96 | 242 | 68% | 357 |
| 0.97 | 225 | 63% | 357 |
| 0.98 | 198 | 55% | 357 |
| 0.99 | 162 | 45% | 357 |
| 0.995 | 133 | 37% | 357 |

# 付録

## Triggerが”Recovery/Exception”である欠陥のstack traceの結果が優れていた

- 全体の欠陥のうち約 11%しか正しく推定できていない
- ”Recovery/Exception”である欠陥については 5つの欠陥のうち 4つを正しく推定できている
  - 全体的な精度に比べると非常に高い精度となった
  - サンプル数は少ないが偶然でない可能性がある

表 10　Trigger が’recovery & exceptions’ であった欠陥に対する stack trace の標準化順位

| Fault | stack trace | dstar | metallaxis | MUSE |
|---|---|---|---|---|
| time-6 | 0.99 | 1.00 | 1.00 | 0.23 |
| chart-17 | 0.99 | 0.99 | 0.99 | 1 |
| lang-6 | 1.00 | 0.25 | 0.80 | 0.02 |
| lang-13 | 0.96 | 0.86 | 0 | 0 |
| lang-14 | 0 | 0.25 | 0 | 0 |