论文汇报

汇报人：陈冰婷　　导师：邹卫琴

Ask the Mutants: Mutating Faulty Programs for Fault Localization

MUSE

MUtation-baSEd fault localization technique, a new fault localization technique basedon mutation analysis.

LIL

Locality Information Los, a new evaluation metric for fault localization techniques based on information theory.

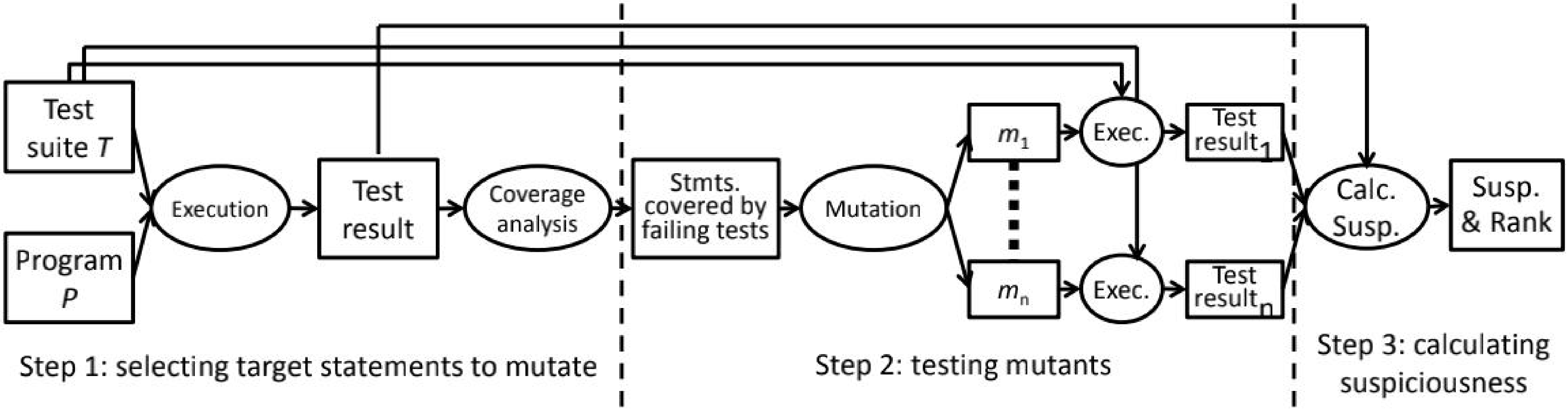

Figure 2: Framework of MUtation-baSEd fault localization technique (MUSE)

MUSE

Example

	`int max;` `void setmax(int x, int y){`	TC1 (3,1)	TC2 (5,-4)	TC3 (0,-4)	TC4 (0,7)	TC5 (-1,3)	$\lvert f_P(s)\rvert$	$\lvert p_P(s)\rvert$	Jaccard Susp.	Jaccard Rank	Ochiai Susp.	Ochiai Rank	Op2 Susp.	Op2 Rank
s_1:	`max = -x; //should be 'max = x;'`	•	•	•	•	•	2	3	0.40	5	0.63	5	1.25	5
s_2:	`if(max < y){`	•	•	•	•	•	2	3	0.40	5	0.63	5	1.25	5
s_3:	`max = y;`	•	•		•	•	2	2	0.50	2	0.71	2	1.50	2
s_4:	`if(x+y<0)`	•	•		•	•	2	2	0.50	2	0.71	2	1.50	2
s_5:	`print(''diff.sign'');}`		•			•	1	1	0.33	6	0.50	6	0.75	6
s_6:	`print(max);}`	•	•	•	•	•	2	3	0.40	5	0.63	5	1.25	5
	Test Results	Fail	Fail	Pass	Pass	Pass								

Statements	Mutants	Test Result Changes TC1 (3,1)	TC2 (5,-4)	TC3 (0,-4)	TC4 (0,7)	TC5 (-1,3)	$\lvert f_P(s) \cap p_m\rvert$	$\lvert p_P(s) \cap f_m\rvert$	MUSE Suspiciousness	Rank
s_1: `max = -x;`	`m1: max -= x-1;`			P→F			0	1	0.46	1
	`m2: max=x;`	F→P	F→P				2	0		
s_2: `if(max < y){`	`m3: if(!(max<y)){`			P→F	P→F	P→F	0	3	0.09	2
	`m4: if(max==y){`	F→P			P→F		1	1		
s_3: `max = y;`	`m5: max = -y;`				P→F	P→F	0	2	-0.16	5
	`m6: max = y+1;`				P→F	P→F	0	2		
s_4: `if(x+y<0){`	`m7:if(!(x+y<0))`				P→F	P→F	0	2	-0.12	4
	`m8:if(x/y<0)`					P→F	0	1		
s_5: `print(''diff.sign'');}`	`m9:return;`					P→F	0	1	-0.08	3
	`m10:;`					P→F	0	1		
s_6: `print(max);}`	`m11:printf(0);}`				P→F	P→F	0	2	-0.20	6
	`m12:;}`			P→F	P→F	P→F	0	3		

Figure 1: Example of how MUSE localizes a fault compared with different fault localization techniques

$$\mu(s) = \frac{1}{|mut(s)|}\sum_{m\in mut(s)}\left(\boxed{\frac{|f_P(s)\cap p_m|}{|f_P|}} - \alpha\cdot\boxed{\frac{|p_P(s)\cap f_m|}{|p_P|}}\right)$$

$$\frac{f2p}{|mut(P)|\cdot|f_P|}\cdot\frac{|mut(P)|\cdot|p_P|}{p2f}$$

Two Conjectures

Conjecture 1: test cases that used to fail on P are more likely to pass on m_f than on m_c.

Conjecture 2: test cases that used to pass on P are more likely to fail on m_c than on m_f.

Why? Expense Metric

$\tau \in [0, 1]$

$$\mathcal{L}(s_i) = \begin{cases} 1 & (s_i = s_f) \\ \epsilon & (0 < \epsilon \ll 1, s_i \in S, s_i \neq s_f) \end{cases}$$

$$P_\tau(s_i) = \frac{\tau(s_i)}{\sum_{i=1}^{n} \tau(s_i)}, (1 \leq i \leq n)$$

$$D_{KL}(P_\mathcal{L} || P_\tau) = \sum_i \ln \frac{P_\mathcal{L}(s_i)}{P_\tau(s_i)} P_\mathcal{L}(s_i)$$

Strengths

1. Expressiveness
2. Flexibility
3. Applicability

未来展望
Future Work

Different Mutation Operators

In-depth study of different mutation operators.

Larger Subject

Apply MUSE to larger subjects such as PHP with multiple test suites.

Other Testing

Apply the mutation idea to conconlic unittesting and concurrent coverage-based testing

THANK YOU FOR YOUR LISTENING.

谢谢您的聆听