

***TERNARIUS:
UM OPERADOR DE MUTAÇÃO
PARA O REPARO DE SOFTWARE
BASEADO EM BUSCA COM
REPRESENTAÇÃO SUBPATCH***

Vinícius Oliveira, Eduardo Souza, Altino Dantas, Lucas Roque,
Celso Camilo-Junior, Jerffeson Souza

Agenda

1. Introdução
2. Background
3. Ternarius
4. Avaliação
5. Considerações Finais

1.

INTRODUÇÃO

CORREÇÃO DE BUGS

70%

Custo do ciclo de vida de um software



Reparo automatizado

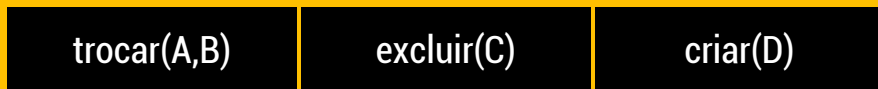
Execução simbólica

Modelos de
aprendizagem de
máquina

Baseado em busca

REPRESENTAÇÃO DE SOLUÇÕES

Uma lista de edições a serem feitas no código original



Uma lista de edições definidas por "subespaços"



**Operador de
Cruzamento**

- OP1Space
- Unif1Space
- OPAlls

**Operador de
Mutação**

?

PRINCIPAIS CONTRIBUIÇÕES

1. Um novo operador de mutação para o reparo de software baseado em busca com representação *subpatch*;
2. Uma avaliação de eficácia da proposta e qualidade dos *patches* gerados.



2.

BACKGROUND

GENPROG



REPRESENTAÇÃO POR PATCH VS SUBPATCH

Patch:

$op_1(fault_1, fix_1)$	$op_2(fault_2, fix_2)$...	$op_n(fault_n, fix_n)$
------------------------	------------------------	-----	------------------------

Subpatch:

op_1	op_2	...	op_n	$fault_1$	$fault_2$...	$fault_n$	fix_1	fix_2	...	fix_n
--------	--------	-----	--------	-----------	-----------	-----	-----------	---------	---------	-----	---------

Operation	Fault	Fix
-----------	-------	-----

EXEMPLO

LIMITAÇÃO DE *PATCH*

Candidato A

trocar(10, 3)

excluir(8)

inserir(5, 12)

Candidato B

inserir(1, 11)

excluir(13)

inserir(8, 4)

Solução aceitável

trocar(10, 3)

inserir(1, 12)

excluir(8)

Inatingível apenas por recombinações

PROPOSTA

Operador de mutação

3.

TERNARIUS

OPERADOR DE MUTAÇÃO

TERNARIUS

- Operador canônico sempre insere uma nova operação no *patch*;
- Operador proposto pode:
 - 1) Deletar operações;
 - 2) Alterar valor de um dos subespaços;
 - 3) Acrescentar novas operações.

EXEMPLO DE APLICAÇÃO DA MUTAÇÃO *TERNARIUS*

swap(s) delete(d) append(a)

Alteração de exclusão

s(8,9)	d(7)	s(5,6)
--------	------	--------

Seleciona e exclui uma operação do *patch*

s(8,9)	d(7)
--------	------

Alteração em subespaço

s(8,9)	d(7)
--------	------

Seleciona uma posição do *patch*

s(8,9)

Seleciona um subespaço e altera seu valor

a(8,9)	d(7)
--------	------

Inserção de nova operação

a(8,9)	d(7)
--------	------

Cria uma nova operação com todos os parâmetros

s(1,3)

a(8,9)	d(7)	s(1,3)
--------	------	--------

A thick, solid red diagonal stripe runs from the top right corner towards the bottom left, separating the white background on the left from a solid red background on the right.

4.

AVALIAÇÃO

QUESTÕES DE PESQUISA

RQ1

O operador Ternarius aplicado à GenProg consegue produzir mais reparos em comparação ao operador canônico?

RQ2

Qual a qualidade dos reparos produzidos pelo operador de mutação Ternarius?

EXPERIMENTOS CONFIGURAÇÕES

- **Benchmark IntroClass;**
 - 5 diferentes programas;
 - Programas com 8, 14 (x3) e 18 versões;
- **GenProg**
 - 30 indivíduos;
 - 30 gerações ou até encontrar reparo;
- **30 execuções para cada versão;**
- **5 variações do Ternarius.**

RESULTADOS

NOVOS OPERADORES (RQ 1)

	ID	Delete	insert	update	Reparos
Mutação canônica	MC	0%	100%	0%	27,962%
	T1	0%	100%	100%	27,654%
	T2	0%	100%	10%	36,358%
Ternarius	T3	10%	100%	0%	29,382%
	T4	80%	100%	0%	34,876%
	T5	80%	100%	10%	26,600%

RESULTADOS

NOVOS OPERADORES (RQ 1)

Programas	Mutação Canônica	T2
checksum	12,92%	17,92% ▲
digits	5,56%	5,56% ▬
median	32,38%	46,19% ▲
smallest	49,76%	70,95% ▲
syllables	5,71%	5,71% ▬

RESULTADOS

AVALIAÇÃO DA QUALIDADE (RQ 2)

Mutação	Qtd. de casos de teste			Qtd. de Reparos			Acurácia
	Exec.	Pos.	Neg.	Encon.	Falsos	Válidos	
MC	3534	3498	36	453	13	440	97,13%
T1	3508	3420	88	448	29	419	93,53%
T2	4518	4404	114	589	45	544	92,36%
T3	3686	3587	99	476	35	441	92,65%
T4	4358	4280	78	565	30	535	94,69%
T5	3372	3296	76	430	24	406	94,42%



5.

CONSIDERAÇÕES FINAIS

CONSIDERAÇÕES FINAIS

- Correção de *bugs* é uma tarefa onerosa;
- Um operador de mutação baseado em representação de subpatch;
- Ternarius produziu mais reparos do que o operador original da GenProg;
- Quantidade vs qualidade dos reparos;
- **Evolução:** utilizar outros datasets, testar combinação com outros operadores de cruzamento.



Obrigado!



altinoneto@inf.ufg.br

Dúvidas?