

Reparo Automatizado de Software

Altino Dantas e Celso G. Camilo-Junior

{altinoneto | celsocamilo}@inf.ufg.br

2018



ALTINO DANTAS

- Cearense de Santa Quitéria, radicado em USS (United States of **Sobral**);
- Bacharel, mestre e **doutorando** em Ciência da Computação;
- Foi membro do GOES.UECE e professor do curso de Sistemas de Informação na Faculdade Lourenço Filho (Fortaleza);
- Publicações nacionais e internacionais em revistas, conferências, revistas, simpósios, inclusive com premiações;
- Atualmente trabalha com Inteligência Artificial, Meta-heurísticas, Computação Evolucionária e Engenharia de Software Baseada em Busca.



O que vem por aí...

- Introdução
- Questões fundamentais
- Principais vertentes
 - ◆ Busca heurística
 - ◆ Modelos de aprendizagem
 - ◆ Execução simbólica
- Considerações finais
- Referências



62 milhões

6,5

Milhões de
linhas de código
em sistemas
aviônicos e
suporte online



2 bilhões



Você realmente acha
que esta situação **é**
sustentável?



Que tal
inspecionar
22k linhas de
código para
encontrar um
bug?!

**Agora
corrija!**

Espere.
Sem
deteriorar
o que está
correto.



↑
Software + ↑
Complexidade

- ▶ A manutenção de software é uma tarefa custosa;
- ▶ Tipicamente **manual** e **complexa**;
- ▶ **Reparo Automatizado de Software.**

“

Reparo automatizado de software consiste em encontrar automaticamente, sem intervenção humana, uma solução para um *bug*. Esta ideia de reparar automaticamente código *bugado* é ao mesmo tempo relevante e desafiadora.



- Martin Monperrus
Automatic Software Repair: a Bibliography

Reparo na prática

```
1 - if (max_range_endpoint < eol_range_start)
2 -     max_range_endpoint = eol_range_start;
3
4 - printable_field = xzalloc(max_range_endpoint/CHAR_BIT+1);
5 + if (max_range_endpoint)
6 +     printable_field = xzalloc(max_range_endpoint/CHAR_BIT+1);
```


Questões fundamentais



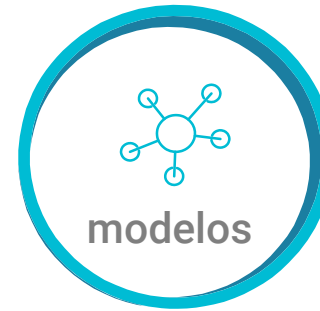
Questões fundamentais

- ▶ Localização do defeito;
- ▶ Magnitude do espaço de busca;
- ▶ Utilização de Casos de Teste;
- ▶ Sintetização e/ou manipulação de código;
- ▶ Qualidade do reparo.

Principais vertentes



Principais vertentes





Técnicas baseadas em **busca**

Características e desafios

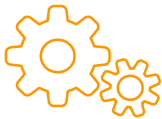
- ▶ Métodos evolucionários;
- ▶ Custo computacional elevado;
- ▶ Manipulação de *patches* ou *AST*;
- ▶ Definição da função objetivo.



busca

Técnicas baseadas em busca

- [2012] - A Systematic Study of Automated Program Repair
Fixing 55 out of 105 Bugs for \$8 Each
- [2017] - Improved representation and genetic operators for
linear genetic programming for automated program repair



busca



Técnicas baseadas em **execução simbólica**

Características e desafios

- ▶ Apoiado por modelagens de restrição;
- ▶ Utilização de *solvers*;
- ▶ Dirigido a casos de teste;
- ▶ Alta qualidade dos reparos;
- ▶ Restrição vs generalização.



Técnicas baseadas em **execução simbólica**

- **[2015]** - Repairing Programs with Semantic Code Search
- **[2016]** - Angelix Scalable Multiline Program Patch Synthesis via Symbolic Analysis





Técnicas baseadas em **modelos de aprendizagem**

Características e desafios

- ▶ Dirigido a “reparos” existentes ou padrões no próprio código;
- ▶ Complexidade e custo computacional elevados;
- ▶ Possibilidade de exploração do conceito de “*Naturalness*”.



modelos



Técnicas baseadas em **modelos de aprendizagem**

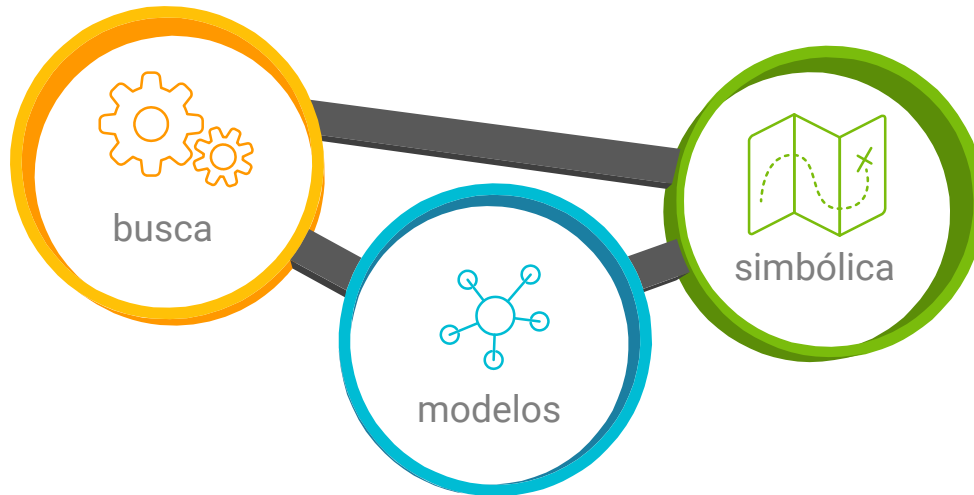
- **[2016]** - Automatic Patch Generation by Learning Correct Code
- **[2017]** - DeepFix: Fixing Common C Language Errors by Deep Learning
- **[2018]** - A new word embedding approach to evaluate potential fixes for automated program repair



modelos

Considerações Finais

Principais vertentes



Referências



- ❖ LONG, Fan; RINARD, Martin. Automatic patch generation by learning correct code. **ACM SIGPLAN Notices**, v. 51, n. 1, p. 298-312, 2016.
- ❖ LE GOUES, Claire et al. A systematic study of automated program repair: Fixing 55 out of 105 bugs for \$8 each. In: **Software Engineering (ICSE), 2012 34th International Conference on**. IEEE, 2012. p. 3-13.
- ❖ OLIVEIRA, Vinicius Paulo L. et al. Improved representation and genetic operators for linear genetic programming for automated program repair. **Empirical Software Engineering**, p. 1-27, 2018.
- ❖ MONPERRUS, Martin. Automatic software repair: a bibliography. **ACM Computing Surveys (CSUR)**, v. 51, n. 1, p. 17, 2018.
- ❖ KE, Yalin et al. Repairing programs with semantic code search (t). In: **Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on**. IEEE, 2015. p. 295-306.
- ❖ MECHTAEV, Sergey; YI, Jooyong; ROYCHOUDHURY, Abhik. Angelix: Scalable multiline program patch synthesis via symbolic analysis. In: **Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on**. IEEE, 2016. p. 691-701.
- ❖ GUPTA, Rahul et al. DeepFix: Fixing Common C Language Errors by Deep Learning. In: **AAAI**. 2017. p. 1345-1351.
- ❖ <https://unsplash.com>.

Obrigado

Dúvidas ou sugestões?

altinoneto@inf.ufg.br

i4soft.com.br