
Modelo para Dissertações em L^AT_EX

Vinícius Humberto Serapilha Durelli

SERVIÇO DE PÓS-GRADUAÇÃO DA UFSJ

Data de Depósito: 10 de Julho de 2019

Assinatura: _____

Modelo para Dissertações em \LaTeX

Vinícius Humberto Serapilha Durelli

Orientador: *Prof. Dr. John Winston Ono Lennon*

Monografia apresentada ao Departamento de Computação (DCOMP) da Universidade Federal de São João del Rei (UFSJ), como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

UFSJ - São João del Rei
Julho/2019

Abstract

HIGH-LEVEL language virtual machines (HLL VMs) have been playing a key role as a mechanism for implementing portable programming languages. Languages that run on these execution environments have many advantages over languages that are compiled to native code. These advantages have led HLL VMs to gain broad acceptance in both academy and industry. However, much of the research in this area has been devoted to boosting the performance of these execution environments. Few efforts have attempted to introduce features that automate or facilitate some software engineering activities, including software testing. This research argues that HLL VMs provide a reasonable basis for building an integrated software testing environment. To this end, two software testing features that build on the characteristics of a Java virtual machine (JVM) were devised. The purpose of the first feature is to automate weak mutation. Augmented with mutation support, the chosen JVM achieved speedups of as much as 89% in comparison to a strong mutation tool. To support the testing of concurrent programs, the second feature is concerned with enabling the deterministic re-execution of Java programs and exploration of new scheduling sequences.

Keywords — software testing; mutation testing; weak mutation; record-and-playback mechanism; Maxine VM; Java virtual machine.

MÁQUINAS virtuais de linguagens de programação têm desempenhado um papel importante como mecanismo para a implementação de linguagens de programação. Linguagens voltadas para esses ambientes de execução possuem várias vantagens em relação às linguagens compiladas. Essas vantagens fizeram com que tais ambientes de execução se tornassem amplamente utilizados pela indústria e academia. Entretanto, a maioria dos estudos nessa área têm se dedicado a aprimorar o desempenho desses ambientes de execução e poucos têm focado o desenvolvimento de funcionalidades que automatizem ou facilitem a condução de atividades de engenharia de software, incluindo atividades de teste de software. Este trabalho apresenta indícios de que máquinas virtuais de linguagens de programação podem apoiar a criação de ambientes de teste de software integrado. Para tal, duas funcionalidades que tiram proveito das características de uma máquina virtual Java foram desenvolvidas. O propósito da primeira funcionalidade é automatizar a condução de atividades de mutação fraca. Após a implementação de tal funcionalidade na máquina virtual Java selecionada, observou-se um desempenho até 89% melhor em relação a uma ferramenta de mutação forte. A fim de apoiar o teste de programas concorrentes, a segunda funcionalidade permite reexecutá-los de forma determinística além de automatizar a exploração de que novas sequências de escalonamento.

Palavras-chave — teste de software; teste de mutação; mutação fraca; mecanismo de record-and-playback; Maxine VM; máquina virtual Java.

Conteúdo

1	Introdução	1
2	Desenvolvimento	3
2.1	Classe PPGCCUFSJ e modelo de trabalho de acadêmico	3
2.2	Codificação dos arquivos: UTF8	4
2.3	Tabelas	4
2.4	Figuras	4
2.5	Diferentes idiomas e hifenizações	5

Lista de Figuras

2.1	Visão geral do pré-processamento de métodos originais antes de serem compilados pela máquina virtual Java.	5
-----	--	---

Lista de Tabelas

2.1	Sumário das características do grupo de tratamento (Durelli et al., 2018). . .	4
-----	--	---

Introdução

Parte inicial do texto, que tem como objetivo elucidar os principais conceitos necessários para compreender os objetivos da pesquisa e outros elementos necessários para apresentar o tema do trabalho (Booth et al., 2008). É importante que a introdução seja clara e convincente.

A equipe de desenvolvimento e manutenção da classe PPGCCUFSJ e do modelo para teses e dissertações em \LaTeX utilizando a classe PPGCCUFSJ é integralmente composta pelas pessoas listadas abaixo. Atualmente, o propósito da equipe é garantir a sustentabilidade deste modelo, tendo autonomia para implementar novos recursos, efetuar compatibilizações necessárias em decorrência de alterações de normas da ABNT e/ou normas e padrões estabelecidos pela comissão de pós-graduação da UFSJ.

Programação

- Vinicius H. S. Durelli – `durelli@ufs.j.edu.br`

Normalização e Padronização

- Vinicius H. S. Durelli – `durelli@ufs.j.edu.br`

O objetivo do presente trabalho é apresentar a classe PPGCCUFSJ e o modelo para teses e dissertações em \LaTeX utilizando a classe PPGCCUFSJ. A expectativa é que a classe PPGCCUFSJ e o modelo proposto auxiliem no aprimoramento da qualidade dos trabalhos acadêmicos produzidos pelos alunos de pós-graduação da UFSJ.

Desenvolvimento

Este capítulo é parte principal da dissertação e deve conter a exposição ordenada e detalhada do assunto. Divide-se em seções e subseções, em conformidade com a abordagem do tema e do método, abrangendo: revisão bibliográfica, materiais e métodos, técnicas utilizadas, resultados obtidos e discussão.

O conteúdo deste documento visa apresentar um tutorial para utilização da classe PPGCCUFSJ, utilizando a estrutura de trabalhos acadêmicos, mas por questões didáticas adotou-se capítulo, seções e subseções diferentes das usualmente utilizadas.

2.1 Classe PPGCCUFSJ e modelo de trabalho de acadêmico

A classe PPGCCUFSJ é uma derivada de um antigo modelo utilizado no Instituto de Ciências Matemáticas e de Computação (ICMC) de São Carlos, Universidade de São Paulo (USP):

O objetivo do projeto é disponibilizar um modelo em \LaTeX para a elaboração de trabalhos acadêmicos (dissertação, trabalho de conclusão de curso (TCC), dentre outros) em conformidade com as normas do departamento de Ciência da Computação (DCOMP) da UFSJ.

Este documento e seu código fonte são exemplos de uso da classe PPGCCUFSJ e do pacote `natbib`.

2.2 Codificação dos arquivos: UTF8

A codificação UTF8 deve ser utilizada para todos os arquivos. Utilize a mesma codificação nos documentos que escrever, incluindo nos arquivos de base bibliográficas `|.bib|`. Para tanto, o arquivo `main.tex` deve conter o seguinte pacote:

```
\usepackage[utf8]{inputenc}
```

2.3 Tabelas

As tabelas e os quadros apresentam os dados de modo resumido, oferecendo uma visão geral do conteúdo em questão, visando facilitar a compreensão do fenômeno em estudo. A diferença básica entre ambas está relacionada ao conteúdo e a formatação. Um exemplo de tabela é mostrado na página seguinte (Tabela 2.1).¹

Tabela 2.1: Sumário das características do grupo de tratamento (Durelli et al., 2018).

Overview of the Treatment Group				
	Google Play Reviews	Rating	Lifespan	Commits
Max	624,366	4.8	2,976	11,049
Min	91	3	759	31
Mean	31,658.4	4.27	1,873.80	2,704.53
Trimmed	7,389.04	4.35	1,895.29	2,318.67
Median	3,523.0	4.4	2,113.5	1,682.5
Std Dev	113,831.38	0.42	641.29	2,748.98
MAD [‡]	4,918.53	0.22	491.48	2,029.68
[‡] MAD stands for median absolute deviation.				

2.4 Figuras

Figuras podem ser criadas diretamente em \LaTeX ou podem ser incorporadas por meio de um arquivo externo, como é o caso da Figura 2.1.² É importante que sejam utilizadas imagens vetoriais no formato PDF. Com isso, o tamanho do arquivo final do trabalho será

¹ A Tabela 2.1 foi extraída de Durelli et al. (2018).

² A Figura 2.1 foi extraída de Durelli et al. (2012).

menor e as imagens terão uma apresentação melhor, principalmente quando impressas, uma vez que imagens vetoriais são perfeitamente escaláveis para qualquer dimensão.

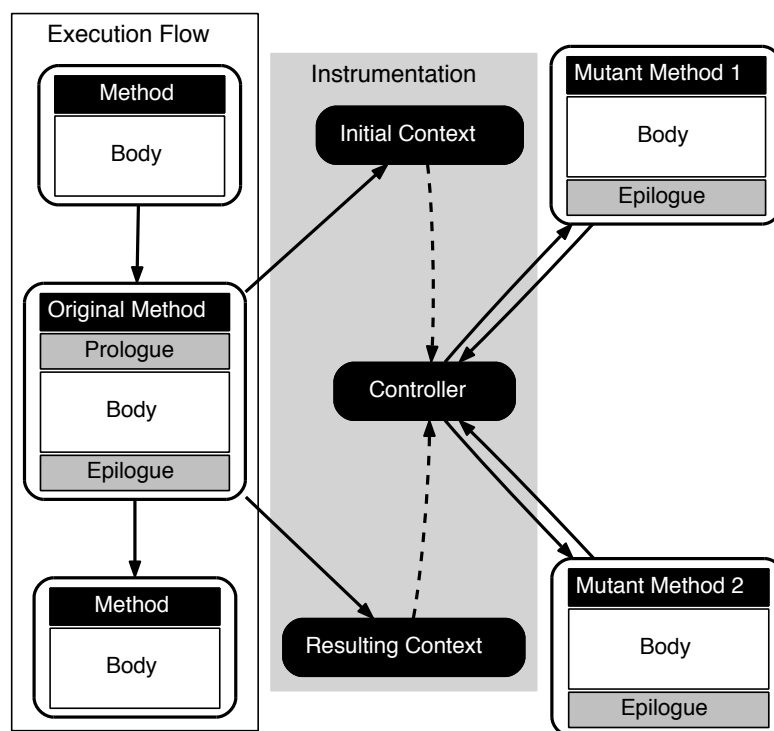


Figura 2.1: Visão geral do pré-processamento de métodos originais antes de serem compilados pela máquina virtual Java.

2.5 Diferentes idiomas e hifenizações

Para usar hifenizações de diferentes idiomas, inclua nas opções do documento o nome dos idiomas que o seu texto contém. Os usuários da classe PPGCCUFSJ devem utilizar:

```
\usepackage[english, %idioma adicional
portuguese]{babel}
```

Desta forma o texto deverá ser escrito idioma português-brasileiro (**brazil**), podendo ter citações em inglês. Os idiomas português-brasileiro (**brazil**) e inglês são incluídos automaticamente pela classe.

Bibliografia

-
-
- BOOTH, W. C.; COLOMB, G. G.; WILLIAMS, J. M. *The Craft of Research (Chicago Guides to Writing, Editing, and Publishing)*. 3rd edition. University Of Chicago Press, 317 pages, 2008.
- DURELLI, V. H. S.; DURELLI, R. S.; ENDO, A. T.; CIRILO, E.; LUIZ, W.; ROCHA, L. Please Please Me: Does the Presence of Test Cases Influence Mobile App Users' Satisfaction? In: *Proceedings of the XXXII Brazilian Symposium on Software Engineering, SBES '18*, ACM, pages 132–141, 2018 (*SBES '18*,).
- DURELLI, V. H. S.; OFFUTT, J.; DELAMARO, M. E. Toward Harnessing High-Level Language Virtual Machines for Further Speeding Up Weak Mutation Testing. In: *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*, pages 681–690, 2012.