

UNIVERSIDADE FEDERAL DE PERNAMBUCO

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

Aplicando regras de programação para refatoração de
programas em Dafny: uma linguagem imperativa com
especificação nativa

Proposta de Trabalho de Graduação

Aluno: Larissa Oliveira Ribeiro da Paz (wtmf@cin.ufpe.br)

Orientador: Márcio Lopes Cornélio (mlc2@cin.ufpe.br)

Recife, 27 de outubro de 2014.

Sumário

Contexto	3
Objetivos	4
Cronograma.....	5
Possíveis avaliadores	6
Referências.....	7
Assinaturas	8

Contexto

Durante a vida útil de um software, frequentemente ocorre a alteração de requisitos, essa alteração comumente implica em mudanças internas no código. Com o amadurecimento desses sistemas há também a demanda de alterações para melhoramento da qualidade do código e de desempenho do sistema. Independentemente da mudança a ser realizada, é preciso bastante cautela para que essas alterações não sejam danosas para o comportamento de outras funcionalidades.

Uma metodologia utilizada para garantir que um sistema funciona de acordo com a especificação é através da programação por contrato (Design by Contract - DbC). Trata-se de uma abordagem de desenvolvimento de software que utiliza métodos formais, por meio da definição pré-condições, pós-condições e invariantes, a fim de facilitar a verificação e garantir a correteza dos sistemas.

Para tentar minimizar os danos decorrentes de alterações no código uma série de procedimentos devem ser empregados. Uma maneira de conduzir este processo é por meio da utilização de leis de programação, que estabelece um suporte para o desenvolvimento formal e rigoroso de sistemas [1]. As leis de programação já foram definidas e adaptadas para os diversos paradigmas de linguagens computacionais [2] [1] [3]. As leis de programação podem ser aplicadas não apenas no código do sistema, mas nos contratos também, para garantir o funcionamento adequado do sistema como todo.

Muitas linguagens vem sendo desenvolvidas com suporte à programação por contrato. Uma delas é Dafny [4,5], uma linguagem de programação que oferece suporte à verificação. Neste sentido, o principal objetivo de Dafny é permitir que programas não tenham erros em tempo de execução e também que funcionem de acordo com a especificação expressa por meio de anotações (contratos). Do ponto de vista de tipos, Dafny é considerada uma linguagem segura. Possui recursos de linguagem de programação imperativa como métodos (procedimentos), comandos de repetição e alocação dinâmica. Além destes, também possui recursos de linguagens de programação funcional (funções recursivas e tipos de dados indutivos e co-indutivos) [4] [5].

Objetivos

O objetivo deste Trabalho de Graduação é investigar como mudanças em código fonte escrito na linguagem Dafny podem ser realizadas de maneira sistemática utilizando leis de programação. Nesta direção, é necessário que, levando em consideração construções presentes em Dafny, as leis de programação sejam adaptadas para Dafny [4,5]. Por exemplo, leis que lidam com contratos de software escritas para a linguagem JML [3] serão avaliadas para adaptação para Dafny. Transformações de programas escritas para linguagens funcionais como Haskell e Erlang [6] [7] serão também avaliadas a fim de determinar quais podem ser adaptadas para Dafny.

Cronograma

A seguir é apresentado o planejamento de execução de atividades durante a produção do Trabalho de Graduação:

	OUTUBRO	NOVEMBRO	DEZEMBRO	JANEIRO
Levantamento Bibliográfico e Definição de Proposta				
Revisão da Literatura				
Adaptação de leis de programação imperativas e de contratos de software (JML)				
Adaptação das regras de refatoração da linguagem funcional				
Proposta de leis de programação e regras de refatoração para Dafny				
Elaboração do relatório				
Elaboração da apresentação				

Tabela 1 - Cronograma

Possíveis avaliadores

Os possíveis avaliadores, na ordem de preferência:

- Alexandre Cabral Mota
- Augusto Sampaio

Referências

- [1] G. F. Freitas, M. Cornélio, T. Massoni e R. Gheyi, “Object-oriented Programming Laws for Annotated Java Programs”.
- [2] P. Borba, A. Sampaio, A. Cavalcanti e M. Cornélio, “Algebraic Reasoning for Object-Oriented Programming,” 2004.
- [3] G. R. F. Freitas, “Refactoring Annotated Java Programs: A Rule-Based Approach,” 2009.
- [4] J. Koenig e K. R. M. Leino, “Getting started with Dafny: a guide,” Marktoberdorf, 2012.
- [5] K. R. M. Leino, “Developing Verified Programs with Dafny,” Redmond.
- [6] G. T. Leavens e Y. Cheon, “Design by Contract with JML,” 2006.
- [7] M. Fowler, “Refactoring,” [Online]. Available: <http://www.refactoring.com/>.
- [8] K. R. M. Leino e R. Middelkoop, “Proving Consistency of Pure Methods and Model Fields”.
- [9] K. R. M. Leino e P. Müller, “Using the Spec# Language, Methodology, and Tools to Write Bug-Free Programs”.
- [10] H. Li, S. Thompson, L. Lövei, Z. Horváth, T. Kozsik, A. Víg e T. Nagy, “Refactoring Erlang Programs”.
- [11] H. Li e S. Thompson, “A Comparative Study of Refactoring Haskell and Erlang Programs”.
- [12] H. Li, C. Reinke e S. Thompson, “Tool Support for Refactoring Functional Programs”.
- [13] H. Li, “Refactoring Haskell Programs,” 2006.

Assinaturas

A estudante e o orientador assinam abaixo se comprometendo com o desenvolvimento do trabalho exposto no presente documento.

Recife, 28 de outubro de 2014.

Márcio Lopes Cornélio (Orientador)

Larissa Oliveira Ribeiro da Paz (Aluno)