XI CONGRESSO DE INICIAÇÃO CIENTÍFICA DA UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

PIBIC/CNPq-UFCG 2014

VERIFICAÇÃO DE IMPLEMENTAÇÕES DE REFATORAMENTO PARA PROGRAMAS NA LINGUAGEM C

Jeanderson Barros Cândido[1], Rohit Gheyi[2]

RESUMO

Refatoramento é o processo de mudar a estrutura de um programa visando melhorar seu design, sem que seu comportamento seja alterado. Desenvolvedores utilizam ferramentas que realizam estas alterações nos seus programas automaticamente. Essas ferramentas verificam um conjunto de condições, porém, não é simples garantir corretude devido à complexidade da semântica das linguagens e ausência de especificações formais. Por tanto, os desenvolvedores de ferramentas de refatoramento costumam implementar os refatoramentos de acordo com suas experiência, utilizando coleções de teste. Porém, não existem evidências que é utilizado um processo sistemático na definição dessas coleções. Neste trabalho, modelamos formalmente um subconjunto da linguagem C e verificamos a implementação de oito refatoramentos do Eclipse CDT. Estes refatoramentos não só modificam a estrutura do programa, como também o corpo das funções, como o refatoramento Extract Function. Detectamos 41 bugs que introduzem erros de compilação no programa resultante, e seis bugs relacionados a mudanças comportamentais. Replicamos manualmente os bugs encontrados em implementações de refatoramentos do NetBeans CND, Xrefactory e OpenRefactory e detectamos 29 bugs nessas ferramentas.

Palavras-chave: teste de software, métodos formais

VERIFICATION OF REFACTORING IMPLEMENTATIONS FOR C PROGRAMS

ABSTRACT

Refactoring is the process of changing the structure of a program to achieve a better design, preserving its behavior. Developers use tools to automatically apply refactorings on their code. These tools check a set of constraints but it is not easy to ensure correctness due to the complex semantics of programming languages and the lack of a formal specification. Therefore, developers of refactoring tools usually implement refactorings guided test suites based on their own experience with the language. However, there is no evidence of having a systematic approach to define such tests. In this work, we created a formal specification for a subset of the C language and evaluated the implementation of eight refactorings of Eclipse CDT. These refactorings not only change the program structure but also the body functions, such as the Extract Function refactoring. We detected 41 bugs that introduce compilation errors and six bugs related to behavioral changes. We analyzed manually these bugs in refactoring implementations of NetBeans, Xrefactory, and OpenRefactory. We identified 29 bugs in these tools.

Keywords: software testing, formal methods

[1]Aluno do Curso de Ciência da Computação, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, e-mail: jeanderson.candido@ccc.ufcg.edu.br

[2]Ciência da Computação, Professor Doutor, Departamento de Sistemas e Computação, UFCG, Campina Grande, PB, e-mail: rohit@dsc.ufcg.edu.br