

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CIN – CENTRO DE INFORMÁTICA

# Uso de espectros de execução para análise de modificações em um programa

Proposta de Trabalho de Graduação

**Aluno:** Rafael Araújo Santana de Oliveira

**Orientador:** Marcelo Bezerra d'Amorim

# 1 Contexto

Espectros de execução são registros provenientes de um programa durante seu funcionamento. Estes registros podem ser gerados através da instrumentação do código do programa para a exibição de informações dos estados percorridos durante a execução do sistema. Comparando espectros de duas versões diferentes de um mesmo programa é possível localizar desvios de comportamento e conseqüentemente os efeitos da modificação entre as versões[1].

Assumindo que da versão original  $v_1$  há uma modificação no código resultando em  $v_2$ , a comparação dos espectros de execução das duas versões do programa nos fornecerá as diferenças no comportamento entre  $v_1$  e  $v_2$ . Se as diferenças entre os espectros podem ser isoladas como consequência direta da modificação de código, então estas alterações são equivalentes semanticamente e provenientes de um *refactoring* correto. Se as diferenças não são consequência direta do código modificado, uma alteração na semântica do programa ocorreu. Esta alteração semântica pode ter sido originada por meio de remoção ou inserção de *bugs* e *features*.

# 2 Aplicação

Um primeiro exemplo demonstrando o uso da comparação de espectros seria o *refactoring* do algoritmo de ordenação *bubblesort*, em  $v_1$ , pelo *quicksort*, em  $v_2$ . Neste caso, as modificações no código, se bem feitas, acarretariam somente no aumento da performance e a semântica se manteria idêntica, já que partindo do mesmo estado o estado final de ordenação é idêntico em ambos algoritmos. As modificações nos espectros seriam diretamente relacionadas à mudança no código entre as duas versões.

Um segundo exemplo seria a correção isolada de um *bug* no algoritmo de ordenação de um programa. Assumindo que nas duas versões logo após a ordenação o programa imprima na tela a lista de itens ordenados, existe alguma lista de itens onde a ordenação é diferente nas duas versões e, portanto, para esta lista o programa apresenta resultados diferentes. Na comparação dos espectros, haveria registro de diferenças no procedimento de impressão dos itens. Como o procedimento de impressão não foi alterado, estas diferenças dos espectros não são consequências diretas da correção e evidenciam mudanças semânticas entre os códigos das duas versões.

Um terceiro exemplo seria a inserção de um *bug* no sistema. Análogo ao segundo exemplo, a análise do espectro revelará diferenças que não são consequências diretas da modificação de código. A comparação do espectro pode auxiliar na procura da falha[2].

### 3 Proposta

Este trabalho se dispõe a analisar o uso do *branch spectra*[1] na classificação de modificações entre duas versões de um sistema. São necessários um instrumentador de código para geração do espectro durante a execução e um comparador de espectros com alguma heurística capaz de inferir se as modificações de comportamento durante a execução são devidas diretamente a alterações no código.

É intenção deste trabalho procurar em sistemas de controle versão por *commits* que evidenciam modificações de código que corrigem *bugs* entre duas versões. Estes sistemas devem ter em seus repositórios de versões, uma suíte de teste capaz de garantir testes de regressão nos bugs corrigidos entre as versões onde ocorre a correção. Esta suíte servirá para classificar se na alteração houve surgimento ou correção de erros de regressão. A heurística deve classificar estas modificações como não relacionadas diretamente a alterações no código.

### 4 Avaliação

O trabalho analisará a qualidade do *branch spectra* para as atividades descritas na proposta. É esperado que a aplicação, utilizando o espectro proposto, seja capaz de indicar se a modificação entre as duas versões é um *refactoring* correto, reportando que  $v_1$  e  $v_2$  tem semântica idêntica. É esperado também que a aplicação, com a ajuda de uma suíte de testes de regressão, seja capaz de indicar que  $v_1$  e  $v_2$  são semanticamente distintos na inserção ou remoção de bugs.

## 5 Cronograma

Atividade	Março	Abril	Maiο	Junho
Lavantamento bibliografico	*	*		
Implementação da solução	*	*	*	
Relatório final			*	*
Apresentação oral				*

## Referências

- [1] Mary Jean Harrold, Gregg Rothermel, Kent Sayre, Rui Wu, and Liu Yi. An empirical investigation of the relationship between spectra differences and regression faults. *Softw. Test., Verif. Reliab.*, 10(3):171–194, 2000.
- [2] Tao Xie and David Notkin. Checking inside the black box: Regression testing by comparing value spectra. *IEEE Trans. Softw. Eng.*, 31(10):869–883, 2005. Member-Xie, Tao and Senior Member-Notkin, David.

## Assinaturas

---

Rafael Araújo Santana de Oliveira

---

Marcelo Bezerra d'Amorim

Recife, 17 de março de 2009