

Marcus Amorim Leal

**Finalizadores e Referências Fracas**  
Interagindo com o Coletor de Lixo

**TESE DE DOUTORADO**

**DEPARTAMENTO DE INFORMÁTICA**  
Programa de Pós-Graduação em  
Informática

Rio de Janeiro  
Setembro de 2005

**Marcus Amorim Leal**

**Finalizadores e Referências Fracas**  
**Interagindo com o Coletor de Lixo**

**Tese de Doutorado**

Tese apresentada ao Programa de Pós-Graduação em Informática do Departamento de Informática da PUC-Rio como requisito parcial para obtenção do título de Doutor em Informática

Orientador: Prof. Roberto Ierusalimsky

Rio de Janeiro  
Setembro de 2005

**Marcus Amorim Leal**

## **Finalizadores e Referências Fracas**

### **Interagindo com o Coletor de Lixo**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Roberto Ierusalimschy**

Orientador

Departamento de Informática – PUC-Rio

**Prof<sup>a</sup>. Noemi de La Rocque Rodriguez**

Departamento de Informática – PUC-Rio

**Prof. Edward Hermann Haeusler**

Departamento de Informática – PUC-Rio

**Prof. Roberto da Silva Bigonha**

Departamento de Ciência da Computação – UFMG

**Prof. Vitor Manuel de Moraes Santos Costa**

COPPE – UFRJ

**Prof. Christiano de Oliveira Braga**

Instituto de Computação – UFF

**Prof. José Eugenio Leal**

Coordenador Setorial do Centro Técnico Científico

Rio de Janeiro, 15 de setembro de 2005

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Marcus Amorim Leal**

Graduou-se em Engenharia de Sistemas na PUC-Rio, obteve o título de Mestre em Administração pelo COPPEAD-UFRJ e o título de Mestre em Informática pela PUC-Rio.

#### Ficha Catalográfica

Leal, Marcus

Finalizadores e referências fracas: interagindo com o coletor de lixo / Marcus Amorim Leal; orientador: Roberto Ierusalimsky. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2005.

v., 140 f: il. ; 29,7 cm

1. Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Tese. 2. Linguagens de programação. 3. Finalizadores. 4. Referências fracas. 5. Coleta de lixo. 6. Semântica. 7. Modelos formais. I. Ierusalimsky, Roberto. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

**Para a minha família.**

## Agradecimentos

Ao Professor Roberto pela valiosa orientação.

À Cecília pela eterna inspiração.

Aos colegas da PUC e do Tecgraf que de alguma forma contribuíram para a realização deste trabalho.

Ao CNPQ pelo suporte financeiro.

## Resumo

Leal, Marcus; Ierusalimschy, Roberto. **Finalizadores e Referências Fracas: Interagindo com o Coletor de Lixo**. Rio de Janeiro, 2005. 140p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Inúmeras linguagens de programação oferecem suporte a finalizadores e referências fracas. Não obstante, de maneira geral esses mecanismos são relativamente pouco conhecidos e pouco usados por programadores. Mesmo entre pesquisadores e desenvolvedores de linguagens não existe muito consenso quanto à sua semântica, que varia consideravelmente entre diferentes implementações. Neste trabalho buscamos explorar os conceitos de finalizadores e de referências fracas, suprindo a ausência de uma especificação clara e abrangente, e permitindo uma melhor compreensão, implementação e uso dos mecanismos correspondentes. Como ponto de partida realizamos um amplo levantamento sobre como é feito o suporte a finalizadores e referências fracas em diferentes linguagens de programação, identificando as características comuns, os problemas, e as questões semânticas mais relevantes associadas às implementações consideradas. Para garantir uma maior precisão em nossa análise, utilizamos um modelo abstrato de uma linguagem de programação com gerenciamento automático de memória. Através deste modelo especificamos formalmente a semântica de finalizadores e referências fracas, incluindo descrições das suas principais variantes e mecanismos relacionados. Além disso, provamos certas propriedades inerentes a linguagens de programação com gerenciamento automático de memória, indicando como estas são afetadas pela introdução de finalizadores e referências fracas. Por fim, consideramos possíveis estratégias de implementação desses mecanismos em diferentes tipos de sistemas. Algumas das opções semânticas investigadas impõe um custo de processamento expressivo, o que frequentemente inviabiliza a sua adoção na prática.

## Palavras-chave

Linguagens de Programação. Finalizadores. Referências Fracas. Coleta de Lixo. Semântica. Modelos Formais.

## Abstract

Leal, Marcus; Ierusalimschy, Roberto. **Finalizers and Weak References: Interfacing with the Garbage Collector**. Rio de Janeiro, 2005. 140p. PhD Thesis — Department of Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Most mainstream programming languages support finalizers and weak references. In spite of that, these abstractions are still modestly known by programmers in general. Even among language designers there seems to be no common view on how to define their semantics, and language implementations certainly reflect that. In this thesis we explore the concepts of finalizer and weak reference by discussing several important issues that, as far as we know, have not been explored by other authors. After presenting a survey on how finalizers and weak references are supported by actual programming languages, we thoroughly examine their semantics and discuss alternative implementation strategies. We also use an operational approach to develop a formal model for reasoning about garbage collection and its interaction with client programs. By explicitly representing low-level details, such as heap memory and its addresses, we were able to clearly specify memory management actions, and prove several important memory-related language invariants. Using this model we describe a formal semantics for finalizers and weak references, exploring some of its many subtleties. We believe that the topics covered here can serve as a relevant reference for further investigations, and also help to guide actual implementations.

## Keywords

Programming Languages. Finalizers. Weak References. Garbage Collection. Semantics. Formal Models.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>12</b>
1.1	Objetivos	15
1.2	Exemplos de Uso	16
1.2.1	Finalizadores	16
1.2.2	Referências Fracas	18
1.3	Organização	20
<b>2</b>	<b>Coleta de Lixo</b>	<b>22</b>
2.1	Contagem de Referências	24
2.2	Mark and Sweep	26
2.3	Mark Compact	26
2.4	Stop and Copy	27
2.5	Coletores Generacionais	28
<b>3</b>	<b>Alternativas de Interface e Suporte</b>	<b>31</b>
3.1	Finalizadores	31
3.1.1	try-finally	31
3.1.2	C++	33
3.1.3	Modula-3	34
3.1.4	Java	35
3.1.5	C#	36
3.1.6	Smalltalk	38
3.1.7	Ruby	38
3.1.8	Python	39
3.1.9	Perl	40
3.1.10	Lua	40
3.2	Referências Fracas	40
3.2.1	Java	40
3.2.2	Eiffel	42

3.2.3	C#	43
3.2.4	Python	43
3.2.5	Smalltalk	44
3.2.6	Perl	44
3.2.7	Lua	44
3.2.8	Haskell	45
3.2.9	Modula-3	46
3.3	Considerações Finais	46
<b>4</b>	<b>Semântica: Uma Análise Informal</b>	<b>47</b>
4.1	Dificuldades de Uso e Implementação	48
4.2	Finalizadores	55
4.2.1	Desacoplamento e Ressurreição	56
4.2.2	Ordem de Invocação	57
4.2.3	Concorrência	59
4.2.4	Sincronismo de Invocação	61
4.3	Referências Fracas	65
4.3.1	Coleta do Objeto Referenciado	65
4.3.2	Desacoplamento e Ressurreição	66
4.3.3	Mecanismos de Notificação	67
4.3.4	Tipos Fracos	68
4.3.5	Coleções Fracas	69
4.4	Referências Fracas e Finalização	71
4.5	Considerações Finais	73
<b>5</b>	<b>Um Modelo Abstrato</b>	<b>75</b>
5.1	A Linguagem de Programação $\lambda_{ref}$	76
5.2	Coleta de Lixo	84
5.3	Finalizadores	89
5.4	Referências Fracas	93
5.4.1	Mecanismos de Notificação	100
5.4.2	Tabelas Fracas	101
5.4.3	Ephemérons	103
5.5	Referências Fracas e Finalização	107
<b>6</b>	<b>Implementação</b>	<b>112</b>
6.1	Coletores de Lixo Baseados em Contagem de Referências	112

<i>Sumário</i>	10
6.1.1 Finalizadores	112
6.1.2 Referências Fracas	115
6.2 Coletores de Lixo Baseados em Rastreamento	116
6.2.1 Finalizadores	117
6.2.2 Referências Fracas	121
<b>7 Conclusão</b>	<b>125</b>
7.1 Contribuições	128
<b>Referências Bibliográficas</b>	<b>130</b>
<b>A Tabelas Fracas</b>	<b>137</b>
A.1 Tabelas Fracas com Referências Fracas	137
A.2 Tabelas Fracas com Ephemerons	139

## Lista de Figuras

2.1	Grafo de conectividade.	23
2.2	Coletor stop-and-copy antes da coleta.	28
2.3	Coletor stop-and-copy depois da coleta.	28
4.1	Uma tabela fraca com valores fortes.	70
5.1	Regras de transição de $\lambda_{ref}$ .	78
5.2	Regras de transição de $\lambda_{ref}$ (cont).	79
5.3	Um grafo de conectividade em $\lambda_{ref}$ .	85
5.4	Rastreando ephemerons: exemplo 1.	106
5.5	Rastreando ephemerons: exemplo 2.	107
6.1	Coleta de lixo e de finalização em coletores mark-and-sweep.	118
6.2	Invocação ordenada de finalizadores com referências cíclicas.	119
6.3	Algoritmo para a coleta e limpeza de ephemerons.	123