

Lucília Camarão de Figueiredo

λ_{Ω} -calculus :

Um Modelo para Não Determinismo em
Linguagens Reflexivas

Tese apresentada ao Departamento de
Ciência da Computação do Instituto de
Ciências Exatas da Universidade Federal
de Minas Gerais, como requisito parcial à
obtenção do título de Doutor em Ciência
da Computação.

Universidade Federal de Minas Gerais
Belo Horizonte, junho de 1997



UFMG - ICEx
DEPARTAMENTO DE CIÊNCIA DA
COMPUTAÇÃO

FOLHA DE APROVAÇÃO

λ_Ω -Calculus: Um Modelo para Não-Determinismo em Linguagens Reflexivas

LUCÍLIA CAMARÃO DE FIGUEIREDO

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

Prof. EDWARD HERMANN HAEUSLER - orientador
Departamento de Informática - PUC-RIO

Prof. ROBERTO DA SILVA BIGONHA - co-orientador
DCC - ICEx - UFMG

Prof. NEWTON JOSÉ VIEIRA
DCC - ICEx - UFMG

Prof. PROF. DALCÍDIO MORAES CLÁUDIO
Instituto de Matemática - PUC-RS

Prof. PAULO AUGUSTO SILVA VELOSO
Departamento de Informática - PUC-RIO

Prof. JOSÉ LUCAS MOURÃO RANGEL NETO
Departamento de Informática - PUC-RIO

Belo Horizonte, 23 de junho de 1997.

Agradecimentos

Primeiramente, gostaria de agradecer ao meu orientador, professor Edward Hermann Haeusler, o seu estímulo, a dedicada e paciente orientação e as diversas sugestões e comentários sobre este trabalho. Não poderia deixar de também agradecer aos professores Roberto Bigonha e Luiz Carlos Pereira que, juntamente com o professor Hermann, tiveram papel fundamental na minha formação em Ciência da Computação. Agradeço, ainda, ao professor Carlos Camarão de Figueiredo, as proveitosas discussões sobre o trabalho e a cuidadosa leitura de versões preliminares deste texto.

Em um nível mais pessoal, gostaria de agradecer aos meus pais e irmãos, o apoio emocional inestimável, sem os quais este trabalho certamente não teria sido concluído. Ao Luiz, meu amado companheiro, que diz merecer uma indenização por ter suportado as minhas ausências, angústias e irritações, ao longo desses três últimos anos, eu agradeço com carinho, e dedico esse trabalho.

Finalmente, agradeço ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - pelo suporte financeiro obtido para o desenvolvimento dessa pesquisa.

Sumário

1	Introdução	1
1.1	Reflexividade	2
1.2	Paralelismo	3
1.3	Reflexividade e Não Determinismo	4
1.4	Tipos	6
1.5	Objetivos do Trabalho	7
1.6	Estrutura do trabalho	7
2	Fundamentos	8
2.1	λ -calculus	8
2.1.1	Sintaxe e notação	9
2.1.2	λ -conversão	9
2.1.3	λ -redução	10
2.1.4	λ -calculus como linguagem de programação	12
2.1.5	Modelos	13
2.2	λ -calculus tipado	16
2.2.1	Sintaxe	16
2.2.2	Conversão e redução	18
2.2.3	Modelos	19
2.3	Tipos dinâmicos	22
3	A linguagem λ_Ω	26
3.1	Operações primitivas	26
3.2	Abstração funcional e aplicação	27
3.3	Sintaxe	29
3.4	Semântica operacional	31
4	λ_Ω-calculus	34
4.1	Ω -conversão	34
4.2	Ω -redução	36
4.2.1	Propriedades	41
5	Modelo semântico para λ_Ω	51

6	λ_{Ω} tipado	61
6.1	Sintaxe	62
6.2	Semântica operacional	64
6.3	Programando em λ_{Ω}	65
6.4	Correção do sistema de tipos	68
6.5	λ_{Ω} -calculus tipado	71
7	Modelo para λ_{Ω} tipado	73
8	Conclusão	80
8.1	Comparação com outros trabalhos	80
8.2	Propostas de extensões ao trabalho	81
	Bibliografia	82

Resumo

Neste trabalho, descrevemos um modelo axiomático para linguagens de programação *reflexivas*, capazes de expressar, de forma natural, a execução em paralelo de programas da própria linguagem. O modelo, que denominamos λ_{Ω} -*calculus*, é obtido pela extensão do λ -calculus de Church com dois operadores meta-linguísticos, que encapsulam as conversões entre programas e suas representações na própria linguagem, além de dois outros operadores primitivos, que implementam a noção de um passo de avaliação de expressões da linguagem.

A consistência do λ_{Ω} -calculus é garantida pela prova da propriedade Church-Rosser para a noção correspondente de ω -redução. Provamos também que a ω -redução satisfaz a propriedade de Normalização.

Uma característica interessante do modelo λ_{Ω} é evidenciar o *não-determinismo* intrínseco a tais linguagens reflexivas, que tem origem na possibilidade de se distinguir na linguagem, de forma explícita, entre *uso* e *menção* de uma expressão. Uma melhor interpretação desse não determinismo é obtida através do modelo semântico apresentado para o cálculo.

Abstract

This thesis presents an axiomatic model for *reflexive* programming languages, which can define, in a simple and natural way, a suitable implementation of parallelism of its own programs. This model, which we call λ_{Ω} -calculus, is obtained as an extension to Church's λ -calculus, by including two metalinguistic *reify* and *reflect* operators, which encapsulate conversions between programs and their representations, and two other primitive operators, which implement the notion of one step evaluation of language expressions.

The consistency of the λ_{Ω} -calculus is guaranteed by the proof of the Church-Rosser property for the corresponding notion of ω -reduction. We also prove that ω -reduction satisfies Normalization.

One interesting feature of the model is to turn evident an intrinsic *nondeterminism* of such reflexive languages. This nondeterminism arises from the possibility to explicitly distinguish in the language between *use* and *mention* of expressions. A suitable interpretation for the nondeterministic expressions of the language is further given by the semantic presented for the λ_{Ω} -calculus.