

Fabiola Fonseca de Oliveira
Orientador: Roberto da Silva Bigonha

Compilação de uma Linguagem Funcional, Orientada por Objetos, para Definição de Semântica Denotacional

Dissertação apresentada ao Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Belo Horizonte
Março de 1998



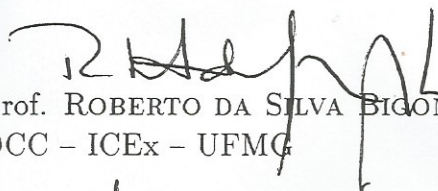
UFMG - ICEx
DEPARTAMENTO DE CIÊNCIA DA
COMPUTAÇÃO

FOLHA DE APROVAÇÃO

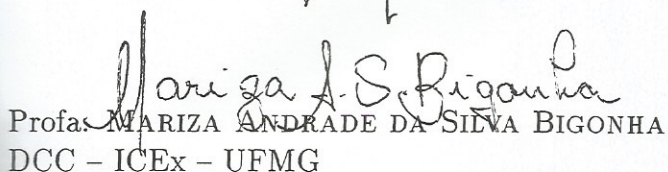
Compilação de uma Linguagem Funcional Orientada por Objetos para Definição de Semântica Denotacional

FABÍOLA FONSECA DE OLIVEIRA

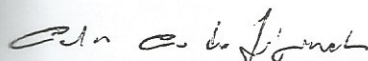
Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:



Prof. ROBERTO DA SILVA BIGONHA – Orientador
DCC – ICEx – UFMG



Profas. MARIZA ANDRADE DA SILVA BIGONHA
DCC – ICEx – UFMG



Prof. CARLOS CAMARÃO DE FIGUEIREDO
DCC – ICEx – UFMG

Belo Horizonte, 31 de março de 1998.

Resumo

O presente trabalho tem como objetivo implementar de forma eficiente um compilador para a linguagem funcional *SCRIPT*, que é utilizada para escrita de definições denotacionais, e que oferece facilidades tais como modularização, equivalência estrutural de tipos e orientação por objetos. Nesta dissertação poderão ser encontradas informações sobre a especificação e a implementação deste compilador, assim como soluções para os problemas encontrados durante seu desenvolvimento, que estão relacionados principalmente com a implementação eficiente de linguagens puramente funcionais com semântica não-estrita, controle de visibilidade, técnicas de importação e exportação entre módulos, compatibilidade e equivalência de domínios e geração eficiente de código *LAMB*.

Abstract

This work focuses on an efficient implementation of a compiler to *SCRIPT*, a functional language for denotational semantics definition that incorporates description modularization, structural type equivalence and object-orientation. In this work, one can find information on the specification and implementation for this compiler and on the solutions for the problems founded during the development of an efficient implementation of purely functional languages with non-strict semantics, control of visibility, export/import facility among modules, compatibility and equivalence of domains, and the code generation into the *LAMB* language.

Agradecimentos

Agradeço ao Prof. Roberto Bigonha pela paciente orientação e permanente estímulo através de sugestões e comentários sobre este trabalho.

Agradeço ao CNPq pelo apoio financeiro, sem o qual este trabalho não teria sido realizado.

Em um nível mais pessoal, gostaria de agradecer a minha família pelo apoio emocional inestimável. Ao Renato, pela paciente compreensão das minhas ausências. Aos amigos do DCC e do convênio DSE, por quebrar a monotonia dos dias de trabalho, principalmente à Jussara, à Erika e ao Yuri, amigos desde a graduação, e à Adriana, à Luisa e ao Guilherme.

A todos, enfim, que de alguma forma contribuíram para a realização deste trabalho.

Sumário

Lista de Figuras	6
1 Introdução	1
1.1 Motivação	1
1.2 Semântica denotacional	2
1.3 Contexto do trabalho	2
1.4 Características importantes	3
1.5 Perspectiva da dissertação	4
2 Revisão da literatura	6
2.1 Cálculo lambda	6
2.2 Técnicas de compilação de linguagens funcionais	7
2.3 Compilação de linguagens funcionais	10
2.4 Conclusão	11
3 A linguagem objeto	12
3.1 A linguagem <i>LAMB</i> utilizada nesta compilação	12
3.2 Descrição da linguagem <i>LAMB</i>	13
3.2.1 Domínios	13
3.2.2 Expressões	13
3.2.3 Padrões	15
3.2.4 Operadores	17
3.3 Conclusão	21
4 A linguagem <i>SCRIPT</i>	23
4.1 Aspectos básicos	24
4.1.1 Palavras-chave	24
4.1.2 Símbolos especiais	24
4.1.3 Identificadores	25
4.1.4 Comentários em <i>SCRIPT</i>	26
4.2 Domínios	26
4.2.1 Domínios padrões	27
4.2.2 Domínios constantes	28
4.2.3 Domínios definidos pelos usuários	28
4.3 Expressões	30
4.3.1 Expressões básicas	30
4.3.2 Expressões de comparação	32
4.3.3 Expressões com padrões	32

4.3.4	Expressões condicionais	33
4.3.5	Expressões de abstração	33
4.3.6	Expressões CASE	34
4.3.7	Expressões LET	35
4.3.8	Atualização de funções	36
4.3.9	Aplicação de funções	36
4.4	Estrutura dos módulos	37
4.4.1	Módulo PROJECT	38
4.4.2	Módulo SYNTAX	38
4.4.3	Módulo MODULE	38
4.5	Conclusão	39
5	O Sistema de tipos <i>SCRIPT</i>	40
5.1	Equivalência e compatibilidade de tipos	41
5.1.1	Equivalência entre D_1 e D_2	41
5.1.2	Compatibilidade entre D_1 e D_2	43
5.2	Disciplina de tipo	44
5.3	Conclusão	56
6	Compilador para <i>SCRIPT</i>	57
6.1	Projeto da compilação	57
6.2	Aspectos básicos	59
6.3	Tabela de símbolos	62
6.3.1	Considerações	62
6.3.2	As estruturas de dados	63
6.3.3	Montando a tabela de símbolos	64
6.4	Análise semântica	65
6.5	Análise semântica Passo I	67
6.5.1	Considerações gerais	67
6.5.2	Atributos	68
6.5.3	Rotinas semânticas	75
6.6	Análise semântica Passo II	87
6.6.1	Análise de dependência	87
6.6.2	Considerações iniciais	87
6.6.3	Definição de atributos	88
6.6.4	Esquemas para a análise de dependência	88
6.7	Análise semântica Passo III	98
6.7.1	Considerações gerais	98
6.7.2	Atributos	98
6.7.3	Rotinas semânticas	103
6.8	Conclusão	117
7	Geração para código <i>LAMB</i>	118
7.1	Considerações iniciais	118
7.2	Compilação de tuplas	118
7.3	Convenções	120
7.4	Esquemas de tradução	120
7.5	Conclusão	132

8	Execução do compilador	133
8.1	Resultados obtidos	133
8.2	Resultados com erros nas declarações	133
8.3	Resultados sobre verificação de equivalência e compatibilidade	134
8.4	Resultados com geração de código <i>LAMB</i>	150
8.4.1	Código <i>LAMB</i> para as funções auxiliares	150
8.4.2	Código <i>LAMB</i> para os programas <i>SCRIPT</i>	152
9	Conclusão	166
	Bibliografia	169
A	A sintaxe de <i>LAMB</i>	173
A.0.3	Palavras chave	176
A.0.4	Símbolos Especiais	177
A.0.5	Expressões Regulares	178
B	A sintaxe de <i>SCRIPT</i>	179
B.1	Definição dos Tokens	179
B.1.1	Palavras chave	180
B.1.2	Símbolos Especiais	182
B.1.3	Expressões Regulares	183
C	Complementação da terceira fase	199
C.0.4	Atributos complementares	200
C.0.5	Rotinas semânticas complementares	207

Lista de Figuras

1.1	Relação entre esta dissertação e os trabalhos de mestrado de Marcelo Maia e de Marco Costa.	3
2.1	Relação entre técnicas para compilação de linguagens funcionais.	10
6.1	Divisão das fases em cada passo.	58
6.2	Aspectos básicos do compilador de <i>SCRIPT</i>	60
6.3	Exemplo da estrutura de dados utilizada para implementar a tabela de símbolos.	66