

N° d'ordre : 272

THESE

présentée devant

l'Université de Rennes I
Institut de Formation Supérieure en Informatique et Communication

pour obtenir

Le Titre de Docteur de l'Université de Rennes I
Mention INFORMATIQUE

par

Pascal FRADET

Titre de la thèse:

COMPILATION DES LANGAGES FONCTIONNELS
PAR
TRANSFORMATION DE PROGRAMMES

Soutenu le 9 Novembre 1988 devant la commission d'examen

<i>MM.:</i>	<i>Laurent</i>	<i>KOTT</i>	<i>Président</i>
	<i>Jean-Pierre</i>	<i>BANATRE</i>	
	<i>Guy</i>	<i>COUSINEAU</i>	<i>Rapporteur</i>
	<i>Jean-Pierre</i>	<i>FINANCE</i>	<i>Rapporteur</i>
	<i>Michel</i>	<i>LEMAITRE</i>	
	<i>Daniel</i>	<i>LE METAYER</i>	
	<i>Philip</i>	<i>WADLER</i>	

Table des matières

Notations

Introduction	1
---------------------------	----------

PARTIE I

LES LANGAGES FONCTIONNELS ET LEURS MISES EN ŒUVRE

Chapitre 1 λ-calcul, combinateurs et langages fonctionnels	5
1.1 Le λ -calcul	6
1.2 La logique combinatoire	10
1.3 Les langages fonctionnels	13
Chapitre 2 Mises en œuvre des langages fonctionnels	15
2.1 Les machines à environnement	16
2.1.1 L'implantation de Lisp	16
2.1.2 La machine SECD	17
2.1.3 La CAM	21
2.2 Les machines à réduction de graphes	26
2.2.1 La machine SK	26
2.2.2 Les supercombinateurs et la G-Machine	29
2.3 Récapitulatif	35

PARTIE II

COMPILATION DES LANGAGES FONCTIONNELS PAR TRANSFORMATION DE PROGRAMMES

Chapitre 3 Motivations et objectif	39
3.1 Expressions fonctionnelles et code machine	39
3.2 Le langage compilé	40
3.3 Objectif et plan	41
Chapitre 4 Compilation du schéma d'évaluation	43
4.1 L'appel par valeur	45
4.1.1 Description de l'appel compilé	45
4.1.2 La transformation \mathcal{V}	47
4.1.3 Propriétés et preuves de correction	50
4.1.4 Application d'une fonction transformée à ses arguments	54
4.1.5 Un exemple de réduction	56
4.2 L'appel par nom	57
4.2.1 Description de l'appel compilé	57
4.2.2 La transformation \mathcal{N}	57
4.2.3 Propriétés et preuves de correction	59
4.2.4 Application d'une fonction transformée à ses arguments	62
4.2.5 Exemple de réduction	63
4.3 Simplifications	64
4.3.1 Mécanismes de base	64
4.3.2 Simplification par évaluation partielle	67
4.3.3 Optimisations des transformations \mathcal{V} et \mathcal{N}	69
4.3.4 Simplifications de l'expression globale	71
4.3.5 Optimisations par analyse de nécessité	73
4.4 Récapitulatif	75

Chapitre 5	Compilation de la gestion des variables	77
5.1	Combinateurs de base	77
5.2	Macro-combinateurs	79
5.3	Normalisation des expressions	80
5.4	L'algorithme d'abstraction	82
5.5	Propriété et preuve de correction	86
5.6	Optimisations de l'algorithme	88
5.7	Optimisations locales	92
5.8	Linéarisation des expressions	96
5.9	Exemple	97
5.10	Récapitulatif	99
Chapitre 6	Traitement des fermetures et appel par nécessité	101
6.1	Introduction d'un argument mémoire	101
6.2	Combinateurs d'accès à la mémoire	102
6.3	La transformation \mathcal{M}	104
6.4	Propriété et preuve de correction	107
6.5	L'appel par nécessité	109
6.6	Optimisations et sujet d'études	111
Chapitre 7	Traduction en assembleur 68020 et résultats	113
7.1	Présentation de la machine et de son assembleur	113
7.2	Introduction de combinateurs spécifiques et pré-traduction	113
7.3	Traduction des expressions en assembleur	115
7.4	Mise en œuvre et résultats	120
Conclusion		123
Bibliographie		127
Annexes		133