



UNIVERSITE DE M'SILA



FACULTE DES MATHÉMATIQUES ET DE
L'INFORMATIQUE

DEPARTEMENT DE MATHÉMATIQUES

MÉMOIRE DE FIN D'ÉTUDE

Présenté pour l'obtention du diplôme de Master

Domaine : Mathématiques et Informatique

Filière : Mathématiques

Option: Mathématiques appliquées et discrètes

Par

Aichaoui Daifa

SUJET

**CALCULABILITÉ ET
FONCTIONS RÉCURSIVES**

Soutenu publiquement le : 23/06/2013

devant le jury composé de :

N. MIDOUNE

Président

M.C/B Université de M'sila

D. MIHOUBI

Rapporteur

Prof. Université de M'sila

N. GHADBANE

Examineur

M.A/A. Université de M'sila

Promotion : 2012 / 2013

4 Fonctions récursives et Modèles de Turing	25
4.1 Définition	25
4.1.1 Description	25
4.1.2 Formalisation	26
4.1.3 Exécution	26
4.1.4 Complexité	27
4.2 Langages acceptés par une machine de Turing	27
4.2.1 Fonctions calculées par une machine de Turing	28
4.2.2 Thèse de Church-Turing. Équivalence M - fonctions récursives partielles	32
Introduction	4
Conclusion	32
1 Notions sur la calculabilité	5
1.1 Définitions formelles de la notion d'algorithme et modèles de calcul	5
1.1.1 Un langage de programmation	6
1.1.2 Quelques exemples	7
2 Fonctions et prédicats primitifs récursifs	9
2.1 Fonctions primitives récursives	9
2.1.1 Réaliser une fonction en un langage	11
2.1.2 Quelques propriétés de fermeture de la classe des fonctions primitives récursives	12
2.1.3 Quelques exemples de fermeture de la classe des fonctions primitives récursives	12
2.2 Prédicats primitifs récursifs	14
2.2.1 Fonction caractéristique d'un prédicat	15
2.2.2 Les fonctions μ récursives	17
2.2.3 Fonctions récursives partielles	17
2.2.4 Fonctions récursives partielles et la thèse de Church	18
2.3 Au-delà des fonctions primitives récursives	18
3 Etude d'une fonction récursive non primitive récursive	20
3.1 Fonction d'Ackermann	20

4 Fonctions récursives et Machines de Turing	25
4.1 Définition	25
4.1.1 Description	25
4.1.2 Formalisation	26
4.1.3 Exécution	26
4.1.4 Configuration	27
4.2 Langage accepté par une machine de Turing	27
4.2.1 Fonctions calculées par une machine de Turing	29
4.2.2 Thèse de Church-Turing. Équivalence M – fonctions récursives partielles	29
Conclusion	31
Bibliographie	32

Le troisième chapitre est consacré à l'étude d'une fonction non primitive réursive, qui est la fonction d'Ackermann.

Le quatrième chapitre nous étudions un autre modèle de calcul qui est la machine de Turing.

Introduction

L'objectif dans ce mémoire Master est l'étude de quelques notions sur la calculabilité que les anglo-saxons désignent par "Theory of computation". Cette théorie s'intéresse généralement aux trois questions fondamentales suivantes:

1. Quels problèmes peut-on (pratiquement) résoudre à l'aide d'un ordinateur ? telle est la question centrale que pose la théorie de calculabilité.
2. Existe-t-il un meilleur algorithme pour un problème démontré soluble? On peut comparer les algorithmes entre eux, on a ce qu'on appelle une hiérarchie de complexité.
3. Existe-t-il un algorithme pour résoudre tel ou tel problème ? c'est la question que pose la théorie de la décidabilité.

Dans ce travail on s'intéresse à l'étude d'un modèle de calcul qui est les fonctions récursives dont l'origine débute depuis les années 1930 par l'étude de la notion de processus effectif de calcul. Différents modèles équivalents pour cette notion ont vu le jour: à l'aide des fonctions récursives par Gödel et Kleene, le λ calcul par Church (1936), les systèmes de post (1936-37) et enfin les Machines de Turing par Turing (1936-37). Dans ce mémoire, on s'intéresse à l'étude des fonctions récursives qui sont obtenu à partir des fonctions de base et les constructeurs : composition, récursion et la minimisation.

Notre travail est organisé en quatre chapitres:

1. Dans le premier chapitre nous donnons quelques notions sur la calculabilité.
2. Dans le deuxième chapitre nous étudions les fonctions et les prédicats primitifs récursifs.

3. Le troisième chapitre est consacré à l'étude d'une fonction non primitive récursive, qui est la fonction d'Ackermann.
4. Dans le quatrième chapitre nous étudions un autre modèle de calcul qui est la machine de Turing.

Chapitre 1

Notions sur la calculabilité

Dans ce chapitre, nous donnons les principaux résultats de la théorie de la calculabilité. La théorie de la calculabilité permet de discuter l'existence, ou la non-existence d'algorithmes pour résoudre un problème donné.

1.1 Définitions formelles de la notion d'algorithme et modèles de calcul

Notre but est de donner une définition formelle de la classe des fonctions calculables. Cette définition doit être à la fois intuitive et correspondre à la pratique de programmation.

Une définition simple est la suivante: une fonction f est calculable s'il existe un algorithme pour la calculer.

L'approche, que nous allons utiliser dans ce chapitre, est dite fonctionnelle. Nous allons reprendre notre attention aux algorithmes pour calculer les fonctions de la forme $f: \mathbb{N}^k \rightarrow \mathbb{N}$, et nous voulons aboutir à une définition (formelle) des fonctions calculables $f: \mathbb{N}^k \rightarrow \mathbb{N}$.

Pour voir le lien entre la décidabilité et la calculabilité, considérons un problème de décision (U, B) où $U = \mathbb{N}^k$. Le problème est donc le suivant: pour $x \in U$ décider si $x \in B$. Afin de décider si une entrée $x \in B$, nous allons utiliser la fonction caractéristique de l'ensemble B $\chi_B(x): U \rightarrow \mathbb{N}$, qui est définie comme suit.

Conclusion

Dans ce mémoire, on a vu l'étude des fonctions récursives qui est un modèle de calcul construit à partir de trois fonctions de bases et les trois opérateurs: composition, récursion et minimisation.

Toute fonction calculable mécaniquement (à l'aide d'un algorithme) peut être obtenue par ces fonctions de bases et ces constructeurs.

On présente aussi un autre modèle plus puissant qui sont les machines de Turing.

- [1] B. MAHMOU, *Langage et complexité*, Cours université Paris 8, 2005.
- [2] CH. BOUTAT, *Récursivité, calculabilité, applications à la théorie des langages*, notes de cours -Grenoble : Université scientifique et médicale, 1983, 1987.
- [3] D. DAVIS, J. WEYERHAESE, *Computability, Complexity, and Languages* Academic Press, 1983.
- [4] D. LASCAN, *Logique mathématique*, tome 2, René Cori, 2011-2012.
- [5] E. ASARI, *Calculabilité*, Cours université de Grenoble, 2005.
- [6] G. A. GILBERT, *Concepts in Theory of Computation and Applications for Automata-Based Machines*, Prentice-Hall, NJ + Brazil, 2005.
- [7] H. ROTTMAN, *Theory of Recursive Functions and Effective Computability*, MIT Press, 1987.
- [8] J. HEIN, *Discrete Structures, Logic, and Computability*, Portland State University, 1998.
- [9] J-P. DELAHAYE, *Calculabilité et décidabilité*, Cours université des Sciences et Technologies de Lille Laboratoire d'Informatique Fondamentale de Lille.
- [10] M. STESSA, *Introduction to the Theory of Computation, Second Edition*, Thomson, 2005.
- [11] P. WOLPER, *Introduction à la calculabilité : Cours et exercices corrigés*, 2e édition, Dunod, 2001.

[13] <http://fr.wikipedia.org/w/index.php?title= Fonction d'Ackermann&oldid=99770581>.

Bibliographie

- [1] B. MARIOU. *Logique et complexité*, Cours university Paris 8, 2005.
- [2] CH. BOITET. *Récurtivité, calculabilité, application à la théorie des langages: notes de cours -Grenoble* : Université scientifique et médicale, 1983, 1987.
- [3] D. DAVIS, J. WEYUKER. *Computability, Complexity, and Languages* Academic Press, 1983.
- [4] D. LASCAR. *Logique mathématique*, tome 2. René Cori, 2011-2012.
- [5] E. ASARIN. *Calculabilité*, Cours université de Grenoble, 2003.
- [6] G A. GIRALDI. *Concepts in Theory of Computation and Applications for Automata-Based Modeling*, Petrópolis, RJ - Brasil, 2005.
- [7] H. ROGERS. *Theory of Recursive Functions and Effective Computability*, MIT Press, 1987.
- [8] J. HEIN. *Discrete Structures, Logic, and Computability*, Portland State University, 1995.
- [9] J-P. DELAHAYE. *Calculabilité et décidabilité*, Cours université des Sciences et Technologies de Lille Laboratoire d'Informatique Fondamentale de Lille.
- [10] M. SIPSER. *Introduction to the Theory of Computation, Second Edition*, Thomson, 2006.
- [11] P. WOLPER. *Introduction à la calculabilité : Cours et exercices corrigés*, 2e édition, Dunod, 2001.

Résumé

Ce travail constitue une introduction à la théorie de la calculabilité .

Le contenu est composé de quatre chapitres. Le premier «Notions sur la calculabilité» présente les notions générales de la calculabilité. Le deuxième «Fonctions et prédicats primitifs récursifs» donne des définitions formelles des fonctions et prédicats primitifs récursifs et on donne quelques exemples.

Le troisième chapitre est un «Étude d'une fonction non primitive récursive» on parle de la fonction d'Ackermann.

Le dernier chapitre «Fonctions récursives et machines de Turing» on utilise un autre modèle du calcul pour étudier les fonctions primitives récursives.

Abstract

This work provides an introduction to the theory of computability.

The content consists of four chapters. The first "Basics computability" presents the notions of computability-general. The second "Functions and primitive recursive predicates" gives formal definitions of functions and primitive recursive predicates and give some examples.

The third chapter is a "Study of a non-primitive recursive function" based on the Ackermann function.

The last chapter "recursive functions and Turing machines" we use another model of computation to study the primitive recursive functions.