

Rapport de Charity

Edwin Ansari

Janvier 2022

1 Introduction

En informatique, un objectif important est d'écrire des programmes corrects et sans bug. Cependant ce but paraît dans un premier temps inaccessible, car la plupart des problèmes admettent des domaines infinis, ce qui rend la validation d'un programme par un nombre de tests impossible. On souhaite donc valider nos programmes de manière plus exhaustive et absolue, ce qui mène naturellement à l'utilisation des outils mathématiques et spécialement la logique mathématique. C'est pourquoi la domaine de spécification et vérification de programme joue un rôle très important en science de logiciel.

D'autre part, un centre d'intérêt en amont des procédures de vérification est le langage de programmation utilisé pour écrire le programme en question. Il est donc possible de faciliter la tâche de validation en étudiant les langages de programmation afin de trouver le langage le plus adapté possible à cette fin. À l'effet de ce dernier des efforts on était mis en œuvre par les spécialistes et concepteurs du domaine, une conséquence générale c'est que les langages de programmation fonctionnel semble être mieux adapté que d'autres paradigmes classiques, dû à leur rapprochement aux mathématiques. Or les constructions de ces langages n'ont qu'un reflet partiel des objets mathématiques et non pas la modélisation exacte de ce dernier. En effet, il est peut-être possible de rendre plus court la transition entre la spécification et l'implémentation d'un programme, en proposant d'autres constructions provenant des théories mathématiques.

Dans ce rapport, on souhaite étudier l'une de ces propositions qui se basent totalement sur la "Théorie des catégories". Un langage proposé est Charity [1] qui a été développé en 1992 à l'Université de Calgary sous la direction de Robin Cockett. Basé sur les idées de Hagino Tatsuya.

References

- [1] Robin Cockett and Tom Fukushima. About charity. Technical report, Citeseer, 1992.