

Учреждение Российской Академии наук
Санкт-Петербургский академический университет –
Научно-образовательный центр нанотехнологий РАН

На правах рукописи

Диссертация допущена к защите
Зав. кафедрой

« » _____ 2014 г.

Диссертация
на соискание ученой степени
магистра

Тема «Экстракция кода из Agda в Haskell»

Направление: 010600.68 — Прикладные математика и физика

Магистерская программа: «Математические и информационные
технологии»

Выполнил студент

Шабалин А. Л.

Руководитель

к.ф.-м.н, доцент

Москвин Д. Н.

Рецензент

???, ???

Малаховски Я. М.

Санкт-Петербург
2014

Содержание

1	Введение	2
1.1	Haskell и Agda	2
1.2	Экстракция кода	2
1.3	Применение экстракции	2
2	Постановка задачи	3
2.1	Цель	3
2.2	Существующие решения	3
2.2.1	Для Coq	3
2.2.2	Для Agda	3
2.3	Анализ MAlonzo	3
2.4	Задачи	3
3	Реализация	4
3.1	Архитектура	4
3.2	TODO: ???	4
4	Заключение	5
4.1	Выводы	5
4.2	Дальнейшая разработка	5
5	Список литературы	6
A	Формальное определение трансформаций	7
B	Доказательство корректности	7

1 Введение

1.1 Haskell и Agda

Haskell¹ — функциональный язык программирования общего назначения.

Agda² — функциональный язык программирования с зависимыми типами и, одновременно, — система компьютерного доказательства теорем.

1.2 Экстракция кода

Термин «экстракция кода» пришел из Coq³ и означает генерацию кода из доказательств[1].

1.3 Применение экстракции

- Техника генерирования верифицированных библиотек — пишем библиотеку с доказательствами на языке с зависимыми типами и генерируем код на языке общего назначения вроде Haskell и ML. Это позволит использовать написанный код из «реальных» приложений, который при этом верифицируется системой доказательства теорем.
- Проще оттранслировать в компилируемый язык, чем написать компилятор. А наличие компилятора позволяет улучшить производительность.

¹<http://haskell.org>

²<http://wiki.portal.chalmers.se/agda/pmwiki.php?n=Main.HomePage>

³<http://coq.inria.fr>

2 Постановка задачи

2.1 Цель

Разработать способ вызывать код, написанный на Agda, из Haskell.

2.2 Существующие решения

2.2.1 Для Coq

Coq очень похож на Agda и поэтому имеет смысл сравнивать их технологии.

Техника называется «экстракция программ»[1]. По программам на Coq генерируются программы на OCaml, Haskell и Scheme. При этом все вычисления, выполняемые только на этапе проверки типов, стираются. То есть, все зависимые типы, и, как следствие, доказательства.

2.2.2 Для Agda

На Agda есть компилятор MAlonzo⁴ (являющийся переписанным компилятором Alonzo[2]), который транслирует код на Agda в код на Haskell и затем компилирует его с помощью ghc, получая в результате исполняемый файл.

2.3 Анализ MAlonzo

TODO: Full description of MAlonzo internals.

2.4 Задачи

1. Реализовать
2. ???
3. PROFIT

TODO: Mention somewhere that the goal is to create a system TODO: that does not break invariants set up by Agda.

⁴<http://thread.gmane.org/gmane.comp.lang.agda/62>

3 Реализация

3.1 Архитектура

3.2 TODO: ???

4 Заключение

4.1 Выводы

4.2 Дальнейшая разработка

5 Список литературы

- [1] P. Letouzey. *A New Extraction for Coq*. TYPES2002, 2002.
- [2] M. Benke. *Alonzo — a compiler for Agda*. TYPES2007, 2007.

A Формальное определение трансформаций

B Доказательство корректности