Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет информационных технологий и программирования
Кафедра компьютерных технологий

Рыбак Андрей Викторович

Представление структур данных индуктивными семействами и доказательства их свойств

Научный руководитель: ассистент кафедры ТП Я. М. Малаховски

Санкт-Петербург
2014

Содержание

Введение

Структуры данных используются в программировании повсеместно для упрощения хранения и обработки данных.

Свойства структур данных происходят из инвариантов, которые эта структура данных соблюдает. Доказать, что структура данных правильно выражена на языке программирования, очень сложно. Практика показывает, что тривиальные структуры и их инварианты данных хорошо выражаются в форме индуктивных семейств. Мы хотим узнать насколько хорошо эта практика работает и для более сложных структур.

Глава 1. Обзор предметной области

В <...> структуры данных позволяют хранить и обрабатывать множество однотипных и/или логически связанных данных в вычислительной технике. Задача (?) структур данных — облегчить написание программ для программистов и ускорить обработку данных.

1.1. Структуры данных

Структуры данных используются в программировании для абстрагирования обработки связанных и однородных данных.

Часто используемые структуры данных ... включаются в стандартные библиотеки языков программирования. Существует несколько различных .... Основные из них:

- foobar

1.1.1. Функциональные структуры данных

... В отличие от ...

1.2. Индуктивные семейства и зависимые типы

Индуктивные семейства — это типы данных, которые могут зависеть от типов и значений. Например, тип векторов индексированных длиной

```
module VecSample where
data ℕ : Set where
  zero : ℕ
  succ : ℕ → ℕ
data Vec A : ℕ → Set where
  nil  : Vec A zero
```

$$\mathsf{cons} : \forall\ \{n\} \to A \to \mathsf{Vec}\ A\ n \to \mathsf{Vec}\ A\ (\mathsf{succ}\ n)$$

Такое определение позволяет нам описать функцию head для такого списка, которая не может бросить исключение:

$$\mathsf{head} : \forall\ \{A\}\ \{n\} \to \mathsf{Vec}\ A\ (\mathsf{succ}\ n) \to A$$

У аргумента функции head тип $\mathsf{Vec}\,A\,(\mathsf{succ}\,n)$, то есть вектор, в котором есть хотя бы один элемент. Это позволяет произвести сопоставление с образцом только по конструктору cons:

$$\mathsf{head}\ (\mathsf{cons}\ a\ as) = a$$

Одной из областей применения индуктивных семейств являются системы интерактивного доказательства теорем.

Индуктивные семейства позволяют формализовывать математические структуры, кодируя утверждения о структурах в них самих, тем самым перенося сложность из доказательств в определения.

1.3. Agda

Agda [1] — чистый функциональный язык программирования с зависимыми типами. В Agda есть индуктивные семейства. В Agda также есть параметризованные модули, mixfix операторы, В коде на Agda широко используются символы Unicode.

В фигурных скобках — неявные аргументы, которые

1.4. Выводы по главе 1

Рассмотрены некоторые существующие подходы к .... Описаны различные .... Кратко описана ... . Кратко описаны особенности языка программирования *Agda*.

Список литературы

1. Agda language. http://wiki.portal.chalmers.se/agda/pmwiki.php.