МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное
учреждение высшего образования
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук
имени И. И. Воровича

Направление подготовки
02.04.02 — Фундаментальная информатика
и информационные технологии

А. И. Насека

РЕАЛИЗАЦИЯ ОБЪЕКТОВ ОБЩЕЙ АЛГЕБРЫ С ПОМОЩЬЮ СИСТЕМЫ COQ

Магистерская диссертация

Научный руководитель:
учёная степень, учёное звание / должность И. О. Фамилия
Рецензент:
ученая степень, ученое звание, должность И. О. Фамилия

Ростов-на-Дону
2019

СОДЕРЖАНИЕ

Введение

Здесь нужно написать введение.

1. Реализация базового множества

1.1. Индуктивное определение множества

Базовое множество будет является начальным объектом при реализации более сложных структур общей алгебры. Поэтому первым объектом формализации будет именно множество.

В системе Coq индуктивное определение множества будет содержать следующую идею: необходимо определить вспомогательную структуру - **список** и наложить на нее ряд **ограничений**. Это можно реализовать разными способами, но для удобства использования был выбран следующий варинт:

```
Variable T : Type.

Inductive BSet : Type :=
    | Empty : BSet
    | Elem : T -> BSet -> BSet
    .
```

Данная структура является классическим списком - он либо может быть пустым (конструктор $Empty$) или состоять из элемента типа T и списка, который в свою очередь также может быть либо пустым, либо содержать элемент и список (конструктор $Elem$). Данное определение является рекурсивным.

Теперь для того, чтобы наш список являлся множеством нам необходимо наложить на него два ограничения (они тоже могут быть сформулированы по разному). Вот реализация данных ограничений в Coq:

- Первое ограничение говорит о том, что, если во множество добавить два одинаковых элемента типа T, то оно не должено измениться. Таким образом данная реализация множества является - **множеством без повторов**. Данное ограничение формализовано в виде аксиомы (использование аксиом в Coq является бо-

лее удобным, чем учитываение этих ограничений сразу в определнии, но позже при реализации более сложных структур будет исползован альтернативный вариант определений):

```
Axiom Ident1 : forall (s : BSet) (x : T),
               Elem x (Elem x s) = Elem x s.
```

- Второе ограничение говорит о том, что порядок добавления элементов во множество не играет никакой роли. Т.е. будем считать множество набором элементов, которые храняться в произвольном порядке. Данное определение также представлено в виде аксиомы:

```
Axiom Ident2 : forall (s : BSet) (x y : T),
               Elem x (Elem y s) = Elem y (Elem x s).
```

Введем обозначения для базового множества с помощью нотации в Coq:

```
Notation "{ # }" := Empty.
Notation "{ # x }" := (Elem x { # }).
Notation "{ # x ; y ; .. ; z }" :=
          (Elem x (Elem y  .. (Elem z { # }) ..)).
```

Данная нотация работает рекурсивно для множества любой длины. После ее введения можно убедиться, что нотация работает корректно, с помощью команды для проверки типов в Coq - $Check$:

```
Check ({ # }).          (* Корректно *)
Check ({# 3}).          (* Корректно *)
Check ({# 3; 4}).       (* Корректно *)
Check ({# 3; 4; 5}).    (* Корректно *)
```

1.2. Базовые операции для множества