**Знания-Онтологии-Теории (ЗОНТ-21)**

# Библиотека теории групп в системе автоматического доказательства Why3

Шелехов В.И.

*Институт Систем Информатики им. А.П. Ершова СО РАН,*
*пр. Лаврентьева, д. 6, г. Новосибирск, 630090, Россия.*
*Новосибирский Государственный Университет*

vshel@iis.nsk.su

**Аннотация.** *Построена библиотека теории групп в системе Why3 на базе стандартной библиотеки теории групп космического агенства NASA в системе доказательства PVS и специальной библиотеки теории групп в системе доказательства Coq. Библиотеку предполагается использовать для быстрого обучения студентов методам автоматического доказательства теорем из области математики в системе Why3. Предлагается новый мини-курс groupWhy3 для обучения студентов университетов методам доказательства теорем на компьютере.*

**Ключевые слова: теория групп, система верификации Why3, дедуктивная верификация, автоматическое доказательство**

## 1 Введение

Автоматическое доказательство теорем на компьютере в основном проводится в целях верификации программ критической инфраструктуры с высокой ценой ошибки. Автоматическое доказательство математических теорем вне производственной верификации случается значительно реже. Тем не менее, число доказанных на компьютере математических теорем исчисляется сотнями. Из них наиболее известным считается доказательство проблемы четырех красок. Примером в теории групп является доказательство в системе Coq [3] теоремы Фейта — Томпсона о разрешимости конечной группы нечётного порядка [6]. Это доказательство проводилось международным коллективом из 15 человек в течение шести лет. Другой пример – доказательство в Coq теоремы Лагранжа: порядок подгруппы конечной группы является делителем порядка конечной группы [5].

Число используемых в мире компьютерных инструментов доказательства исчисляется десятками. Ни одна из этих систем автоматического доказательства теорем пока еще не может применяться в качестве полноценного рабочего инструмента для профессионального математика. Однако через пять-семь лет с появлением нового поколения систем доказательства ситуация может кардинально поменяться, и успех в работе будет зависеть от уровня владения подобными инструментами. С учетом такой перспективы необходимо освоение инструментов автоматического доказательства профессиональными математиками уже в настоящее время. Актуальным становится создание в университетах новых курсов освоения инструментов автоматического доказательства.

В данной статье описывается опыт разработки библиотеки теории групп в системе Why3 [1]. Библиотеку предполагается использовать для быстрого обучения студентов методам автоматического доказательства в системе Why3. Доказательство лемм из этой библиотеки студентами будет их индивидуальным заданием по курсу обучения.

Наша библиотека теории групп для системы Why3 построена на базе стандартной библиотеки теории групп космического агенства NASA для системы автоматического доказательства PVS и специальной библиотеки теории групп [7] для системы доказательства Coq.

На текущий момент времени система доказательства Why3 является наиболее подходящей для обучения среди других популярных систем. Она проста, достаточно универсальна и вполне эффективна за счет привлечения других систем автоматического доказательства.

# 2 Система верификации Why3

Система Why3 [1] является платформой для дедуктивной верификации программ на языках Си, Java и Ада, транслируемых на функциональный язык WhyML, являющийся базовым в системе Why3. Функции на языке WhyML снабжаются спецификациями в виде предусловий и постусловий. По программе и спецификациям система Why3 на базе классической логики Хоара [8] генерирует формулы корректности, доказательство которых гарантирует корректность программы относительно спецификаций.

Доказательство формул корректности реализуется с возможностью привлечения до двадцати внешних инструментов доказательства, автоматических и интерактивных. Запуск автоматического инструмента либо доказывает формулу корректности, либо завершается без результата по истечению определенного фиксированного времени. При запуске одного из интерактивных инструментов: Coq [3], PVS или Isabelle/HOL – пользователь самостоятельно строит доказательство с помощью команд инструмента в диалоговом режиме.

Кроме того, Why3 имеет свои собственные средства интерактивного доказательства, позволяющие декомпозировать и упрощать доказываемую формулу до состояния, когда ее можно будет доказать применением автоматических инструментов. Эти интерактивные средства стали полноценными лишь в последние два года. Ранее возможности интерактивного доказательства были слабые, что часто вынуждало переводить доказательство в Coq [9, 10].

Инструмент Why3 можно применять не только как систему верификации. Инструмент Why3 можно также использовать в качестве системы доказательства теорем для произвольных теорий, записанных на языке формул, являющемся частью языка WhyML. Именно в таком стиле ранее использовалась система Why3 для дедуктивной верификации предикатных программ: формулы корректности предикатных программ переводились на язык формул WhyML и доказывались в системе Why3 [9-11]. Аналогичным образом используется система Why3 при построении базисной библиотеки теории групп.

Система доказательства Why3 [1] намного проще известных систем интерактивного доказательства Coq и PVS. Система доказательства Why3 работает в логике первого порядка с непараметрическими типами, поскольку вызываемые автоматические инструменты автоматического доказательства базируются на логике первого порядка. Тем не менее, система Why3 универсальна и потенциально применима для доказательства любых математических теорем. Создатели системы Why3 много сделали для достижения ее универсальности. В частности, простые функции могут быть параметрами формул.

Программа на языке WhyML состоит из нескольких модулей, связанных между собой отношением импорта. Каждый модуль определяет набор типов, констант, предикатов, функций, аксиом, теорем и целей. Новый предикат вводится описанием вида:

predicate <имя предиката>(<параметры>) = <формула>

Функция определяется следующим синтаксисом:

function <имя функции>(<параметры>)<спецификация> = <тело функции>

Тело функции определяет исполняемую программу функции. Тело функции и/или спецификация могут отсутствовать. Функция, определяемая спецификацией без тела, может использоваться только в других спецификациях. Цель – это формула (теорема), истинность которой должна быть доказана в системе Why3. Теорема (лемма) – это формула, которая должна быть доказана. Теорема может использоваться для доказательства других целей и теорем ниже по тексту модуля.

Спецификация функции определяется следующими конструкциями для предусловия и постусловия:

```
requires { <формула для предусловия> }
ensures { <формула для постусловия>}
```

Формула для предусловия определяет ограничения на значения параметров функции. Формула для постусловия связывает значения аргументов и результата функции, представленного в формуле стандартным идентификатором result.

При наличии спецификации и тела функции генерируются формулы корректности тела функции относительно спецификации. Эти формулы должны быть доказаны. Если функция представлена спецификацией без тела функции, то формулы для предусловия и постусловия используются в качестве аксиом функции.

В состав системы Why3 входит стандартная библиотека теорий (модулей), состоящая из более чем 40 отдельных библиотек. Фактически, стандартная библиотека является продолжением языка WhyML. В частности, тип int определяется в библиотеке int через коммутативное кольцо, определенное в библиотеке algebra, кольцо определяется через группу, а группа через моноид в той же библиотеке algebra. Таким образом, корневой модуль algebra.Group для нашей библиотеки теории групп находится в стандартной библиотеке.

# 3 Построение библиотеки теории групп

Библиотека теории групп [4] построена как продолжение стандартной библиотеки Why3. Подробное описание библиотеки теории групп дано в руководстве [2]. В стандартной библиотеке Why3 группа определена в библиотеке algebra модулем Group:

```
module Group
  clone export Monoid with axiom Assoc, axiom Unit_def_l, axiom Unit_def_r
  function inv t : t
  axiom Inv_def_l : forall x: t. op (inv x) x = unit
  axiom Inv_def_r : forall x: t. op x (inv x) = unit
end
```

Конструкция clone export Monoid реализует копирование модуля Monoid внутрь модуля Group. Аналогичным образом, Monoid копирует модуль Assoc, в котором тип t и бинарная операция op определяются как неинтерпретированные следующими конструкциями:

```
type t
function op  t  t : t
```

Здесь определен произвольный тип t и произвольная бинарная операция op.

Таким образом, тип t определяет множество элементов группы с бинарной операцией op, константой unit – единицей группы, унарной операцией inv взятия обратного элемента и пятью аксиомами, причем три аксиомы переносятся копированием.

Головным модулем библиотеки теории групп является Group_def. Он приведен ниже. Вводятся другие обозначения: e для единицы группы вместо unit и инфиксная операция * вместо бинарной операции op. Это реализуется клонированием (копированием) модуля algebra.Group, причем наследуются все аксиомы. Теперь op x y записывается как x * y.

```
module Group_def
  use int.Int
  type t
  constant e : t
  function (*) t t : t
  function inv  t : t
  clone export algebra.Group   with type t = t, constant unit = e,
                        function op = (*), function inv = inv,  axiom .
end (*Group_def*)
```

Библиотека теории групп построена на базе стандартной библиотеки теории групп космического агенства NASA версии 5.7.3 (в директории algebra) для системы автоматического доказательства PVS и специальной библиотеки теории групп для системы доказательства Coq версии 8.10.0 [7]. Библиотека NASA для PVS включена практически полностью. Из специальной библиотеки для Coq включено все, кроме произведения групп.

Вслед за головным модулем следует модуль Group_lems с 15-ю леммами для операций * и inv. Все они доказываются автоматическими решателями. Далее следует модуль Exponential, определяющий инфиксную операцию ^ возведения в степень. Операция x ^ n возведения элемента группы x в целую степень n определяется как n-кратное умножение x * x * ... * x . Если n – отрицательное, то операция ^ применяется к элементу inv x. Четыре леммы из шести не доказываются автоматическими инструментами, поскольку требуют доказательства по индукции. Модуль Expt_lems содержит 19 лемм для операции ^. Все леммы, кроме одной, доказаны автоматическими инструментами.

Показательно, что все леммы модулей Group_lems и Expt_lems, кроме одной, доказаны автоматическими инструментами. Доказательство этих простых лемм в системах PVS и Coq потребует от трех до пяти дней работы. Это реально подтверждает эффективность системы Why3.

В библиотеке теории групп множество элементов группы представлено неинтерпретированным типом t. Далее при определении подгрупп и конечности групп возникают проблемы. Оказывается, подобного рода проблемы возникают также в системах PVS и Coq, поэтому группы там изначально определяется на множествах. В нашем случае необходимо будет распространить определение группы на множества. Понятие группы далее доопределяется для произвольных подмножеств исходного типа t.

Множества и операции с ними определены в библиотеке set системы Why3. Тип множества подмножеств в языке WhyML определяется конструкцией: set u, где u – некоторый базисный тип. Рассмотрим подмножество g типа t. Множество g имеет тип set t. Модуль GroupSub библиотеки теории групп определяет предикат group g следующим образом: операции * и inv должны быть замкнуты относительно подмножества g. Леммы для групп-множеств представлены модулем GroupSub_lem. Модуль FiniteSet содержит необходимые расширения библиотеки set .

Модуль GroupGen определяет построение минимальной подгруппы, содержащей исходное подмножество s, посредством замыкания относительно операций * и inv. Этому модулю нет аналогов. В библиотеке NASA для PVS определяется лишь замыкание одноэлементного множества относительно операции ^. Леммы находятся в модуле GroupGen_lems.

# 4 Преподавание методов доказательства на компьютере

В настоящее время система доказательства Why3 является наиболее подходящей среди других популярных систем для обучения студентов университетов методам автоматического доказательства математических теорем на компьютере.

Предлагаемый курс обучения groupWhy3 состоит из одной лекции, одного практического занятия и индивидуального задания для студентов. В лекции после краткого введения определяются язык формул и базисные типы данных функционального языка WhyML, излагается механизм действия основных команд доказательства формул и проводится экскурс по стандартной библиотеке. На практическом занятии показывается полный цикл работы в системе Why3. На примере трех лемм из

библиотеки теории групп показываются методы интерактивного доказательства применением разных команд системы Why3.

Далее каждый студент выбирает одну или две леммы из недоказанных в библиотеке теории групп для индивидуального задания. Студентам необходимо установить систему Why3 на свой компьютер. Дистрибутив для Windows с инструкцией: https://persons.iis.nsk.su/files/persons/fmcfiles/why3cdecl.pdf. В дистрибутиве системы Why3 подключены шесть инструментов доказательства: Alt-Ergo 2.4.0, CVC4 1.7, Z3 4.8.6, CVC3 2.4.1 и Vampire 4.2.2, а также Coq 8.12.2.

Результат выполнения задания в формате системы Why3 студент посылает преподавателю. Преподаватель в системе Why3 на своем компьютере оценивает объем и качество проведенных доказательств.

В настоящее время в библиотеке теории групп 14 недоказанных лемм. В дальнейшем библиотека будет расширяться новыми модулями с наборами теорем и лемм.

Мини-курс groupWhy3 может быть самостоятельным курсом в программе обучения бакалавриата. Однако более целесообразно преподавать его в составе курса по теории групп или в составе курса по алгебре, включающего теорию групп.

Описанная схема преподавания обкатана на другом курсе. Исходным базисом предлагаемого курса groupWhy3 является подкурс "Язык спецификаций и методы автоматического доказательства в системе Why3", который второй год читается на ММФ НГУ в магистратуре в составе годового курса «Формальные методы в программной инженерии». В подкурсе две лекции и практическое занятие. Он значительно сложнее предлагаемого курса groupWhy3. Имеются видеолекции подкурса.

Отметим, что курс groupWhy3 может читаться в любом университете России. Он значительно проще любой части курса «Формальные методы в программной инженерии», который по силам лишь университетам первой десятки рейтинга.

Система доказательства Coq [3] весьма популярна в сфере образования. Однако система Coq очень сложна. Доказательство в системе Coq является сложным и требует много времени. В 2013 в МГУ на кафедре логики появилось краткое учебное пособие по Coq. Имеется десять полномасштабных видеолекций Антона Трунова по системе Coq в составе курса по верификации программ [12], прочитанного в МФТИ в 2020г.

Очевидно, что в плане начального обучения методам доказательства на компьютере система Why3 выглядит намного привлекательней системы Coq.

# 5 Заключение

Дальнейшая задача – внедрение курса groupWhy3 в НГУ и в других университетах России и развитие библиотеки теории групп объединенными усилиями при поддержке одного из Российских фондов.

В дальнейшем целесообразно предложить начальную часть библиотеки теории групп для включения в стандартную библиотеку системы Why3. С этой целью необходимо будет провести оптимизацию библиотеки. Наверняка потребуются консультации от разработчиков системы Why3. В частности, возможно следует перенести четыре леммы из модуля Exponential в модуль Expt_lems.

Интригующий вопрос: возможно ли эффективное использование библиотеки теории групп для доказательства актуальных теорем из теории групп в ближайшие три года. Потенциально, система Why3 способна полностью автоматически, применением лишь автоматических инструментов доказательства, доказать любую теорему при условии, если снабдить ее соответствующими промежуточными леммами. Сформулировать такие промежуточные леммы – сложнейшая задача. Подобная задача возникает при доказательстве по индукции. Лучшие инструменты, такие как CVC4 и Z3, способны автоматически доказывать по индукции, однако редко достигают успеха. Кстати, именно по этой причине не доказаны автоматически многие леммы из библиотеки теории групп. Проблема автоматического доказательства по индукции успешно решается, если использовать подходящую промежуточную лемму. Однако определить такую лемму весьма сложно. В дедуктивной верификации для этого используют метод лемма-функций. В реальной практике доказательства в

системе Why3 применяется интерактивное доказательство, в ходе которого явным образом определяются промежуточные леммы, необходимые для завершения доказательства.

## Литература


[1] Why 3. Where Programs Meet Provers. URL: http://why3.lri.fr

[2] Описание библиотеки теории групп в системе доказательства Why3. https://persons.iis.nsk.su/files/persons/pages/group_why3_tut.pdf

[3] The Coq Proof Assistant. https://coq.inria.fr/

[4] Библиотека теории групп в системе Why3 // ИСИ СО РАН, Новосибирск, 2021. https://persons.iis.nsk.su/files/persons/pages/group1.zip

[5] Dave King. Formalizing Lagrange's Theorem in Coq. 2019. https://www.tildedave.com/2019/07/18/formalizing-lagranges-theorem-in-coq.html

[6] Gonthier G., et al., "A Machine-Checked Proof of the Odd Order Theorem" // Interactive Theorem Proving. LNC S 7998, 2013, pp. 163–179. https://hal.inria.fr/hal-00816699/document

[7] Coq-group-theory 8.10.0. Elements of Group Theory. http://coq.io/opam/coq-group-theory.8.10.0.html

[8] Hoare C. A. R.: An axiomatic basis for computer programming // Communications of the ACM. 1969. Vol. 12 (10). pp. 576–585.

[9] Шелехов В.И. Верификация предикатной программы пирамидальной сортировки с применением обратных трансформаций // Системная информатика, № 16. Новосибирск, 2020. С. 75-102.

[10] Шелехов В.И. Верификация программы преобразования строки в целое число // Системная информатика, № 17. — Новосибирск, 2020. — С. 43-90. https://persons.iis.nsk.su/files/persons/pages/kstr2.pdf

[11] Тумуров Э.Г., Шелехов В.И. Трансформация, спецификация и верификация программы вычисления числа элементов множества, представленного в виде битовой шкалы. — Системная информатика, № 16. — Новосибирск, 2020. — С. 103-136.

[12]Трунов Антон. Введение в верификацию программ. Видеолекции курса. 2020. https://www.youtube.com/playlist?list=PLQ XJe6nVU5BwX1gV80aDhwW0Jt3tipIY