МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ярославский государственный университет им. П.Г.Демидова»

Кафедра компьютерной безопасности и математических методов
обработки информации

Курсовая работа

Алгоритм Тарского. Описание и реализация

Научный руководитель
профессор, д-р ф.-м.н.
________________ В. Г. Дурнев
«___» _________ 2020 г.

Студент группы КБ-41СО
________________ Р. А. Гибадулин
«___» _________ 2020 г.

Ярославль, 2020 г.

Оглавление

Введение **2**

1 Алгоритм Тарского **3**
1.1 Элементарная алгебра . . . 3
1.2 Язык элементарной алгебры . . . 3
1.3 Поле действительных чисел . . . 3
1.4 Элиминация кванторов . . . 3
1.5 «Алгоритм» Тарского . . . 4
1.6 Алгоритм Тарского . . . 4

2 Программная реализация **6**
2.1 Представление формул. Библиотека LogicLanguageLib . . . 6
2.2 Система ввода. Класс Parser . . . 6
2.3 Реализация базового случая. Библиотека SimpleTarskiAlgorithmLib . . . 6
2.4 Библиотека SimpleTarskiAlgorithmRunner . . . 7
2.5 Юнит-тестирование . . . 7
2.6 Результаты работы . . . 7
2.7 Общий случай . . . 7

Заключение **8**

Список литературы **9**

Приложение А **10**

Введение

Введение

1 Алгоритм Тарского

вступление

1.1 Элементарная алгебра

Неформальное описание элементарной алгебры. Смотри первоисточник и википедия.

1.2 Язык элементарной алгебры

Формальное описание языка: Ссылка на учебник Дурнева. Сигнатура языка и алгебраическая система. Обратить внимание на $\mathbb{R}$. Ввести понятие нуль-местного предиката. Примеры утверждений в этом языке. Замечания по про: перенос в неравенствах, про дроби, про коэффициенты многочленов (что они из Q и вообще хватит только 0, 1, -1), про операции – хватит $+$ и *, про нульместный предикат, про то что хватит $<$ и $=$,

определение: простой формулой будем называть формулу, в которой любая подформула вида QxA не содержит свободных вхождений переменных.

Определение 1.1. *Формула $\mathcal{A}$ языка элементарной алгебры называется **простой**, если все подформулы формулы $\mathcal{A}$ вида $(Qx)\mathcal{B}$, где Q – квантор, не содержат свободных вхождений переменных.*

1.3 Поле действительных чисел

Теоремы из матана и алгебры про многочлены (ссылки на соответствующие книги), нули монотонной функции, монотонность многочленов, расположение корней производной (картинка нужна), значение многочлена в точке равно значению остатка в точке Система из нескольких многочленов. Производная произведения.

1.4 Элиминация кванторов

Элиминация кванторов определение. Рассуждение – хотим проверять истинность формул. Умеем это делать в логике высказываний. Значит "избавившись от кванторов"мы сведем задачу к задаче проверки истинности утверждения в логике высказываний. Пусть алгоритм элиминации существуют, тогда: общая схема индуктивного "алгоритма"для любого языка – нет кванторов – задача решена, один квантор – применяем алгоритм элиминации, получаем бескванторную, несколько кванторов – элиминируем "внутренний получили n-1 квантор, задача решена.

1

2

3

4

5

6

7

8

9

Пусть алгоритм A такой, что $A\left(\left(Qx\right)\mathcal{A}\right) = \mathcal{B}$, где $\mathcal{A}$ и $\mathcal{B}$ – бескванторные формулы языка элементарной алгебры, и формулы $(Qx)\mathcal{A}$ и $\mathcal{B}$ эквивалентны. Тогда выполнено следующее утверждение:

Утверждение 1.1. *Если алгоритм A существует, то существует алгоритм B такой, что для любой формулы $\mathcal{A}$ языка элементарной алгебры $B\left(\mathcal{A}\right)$ – бескванторная формула, эквивалентная $\mathcal{A}$.*

Доказательство. Определим алгоритм B следующим образом:

- Если $\mathcal{A}$ – бескванторная формула, то $B\left(\mathcal{A}\right) = \mathcal{A}$;
- Если $\mathcal{A} = \left(Qx\right)\mathcal{B}$, то $B\left(\mathcal{A}\right) = A\left(\left(Qx\right)B\left(\mathcal{B}\right)\right)$. Формула $B\left(\mathcal{B}\right)$ – бескванторная по построению B, следовательно запись $A\left(\left(Qx\right)B\left(\mathcal{B}\right)\right)$ корректна, при этом $B\left(\mathcal{B}\right)$ эквивалентна $\mathcal{B}$, следовательно, $\left(Qx\right)\mathcal{B}$ эквивалентна $\left(Qx\right)B\left(\mathcal{B}\right)$, а значит $\mathcal{A}$ эквивалентна $B\left(\mathcal{A}\right)$. Также заметим, что длина формулы $\mathcal{B}$ строго меньше длины формулы $\mathcal{A}$;
- Если $\mathcal{A}$ не удовлетворяет предыдущим условиям, то
  - либо $\mathcal{A} = \neg\mathcal{B}$, тогда $B\left(\mathcal{A}\right) = \neg B\left(\mathcal{B}\right)$,
  - либо $\mathcal{A} = \mathcal{B} * \mathcal{C}$, тогда $B\left(\mathcal{A}\right) = B\left(\mathcal{B}\right) * B\left(\mathcal{C}\right)$, где $* \in \{\vee, \&, \to\}$.

  При этом длины формул $\mathcal{B}$ и $\mathcal{C}$ меньше длины формулы $\mathcal{A}$.

Алгоритм B определен рекурсивно, при этом на каждом этапе на вход B подаётся формула меньшей длины, следовательно, алгоритм B является конечным, и на каждом шаге выход алгоритма – бескванторная эквивалентная формула. ◀

Замечание. *Данное утверждение верно и для других языков логики предикатов.*

Таким образом, для элиминации кванторов в произвольной формуле достаточно построить алгоритм A.

1.5 «Алгоритм» Тарского

Изначально предложенный Тарским «решающий метод» (decision method (ссылка на работу Тарского)) имел другой вид, но Тарский был первым. А алгоритм, который приведен в данной работе, был предложен уже «последователями» Тарского, однако в литературе (отсылка на Матиясевича) именно этот алгоритм носит имя Тарского.

Описываем алгоритм, в котором есть слова "ищем корни многочленов"сформулировать список того, почему это не настоящий алгоритм.

1.6 Алгоритм Тарского

Сначала рассмотрим базовый случай

Напоминаем про производную произведения. Что такое таблица Тарского. Сокращенная таблица Тарского. Свойства Таблицы Тарского. Насыщенная система. Алгоритм построения таблицы Тарского. Окончательное описание алгоритма.

Теперь перейдем с общему случаю

Особенности общего случая. Над каким полем многочлены. Описание алгоритма: ветвление в листах базовый случай

Теорема Тарского (2 формулировки)

Попробовать доказать корректность алгоритма, воспользовавшись утверждением из википедии про упрощение формулы.

Оценки: количество многочленов, количество многочленов в насыщенной системе, размеры таблицы, количество ветвлений.

МИНИЗАКЛЮЧЕНИЕ: существование алгоритма считалось невозможным. Давайте реализуем его.

2 Программная реализация

Для реализации программы, которая по простой формуле (определение 1.1) языка элементарной алгебры, вводимой пользователем с клавиатуры, строит эквивалентную бескванторную формулу, были выбраны язык C#, платформа .NET Core 3.1 и спецификация .NET Standard 2.1 [4]. Такой выбор обусловлен рядом причин:

- Язык C# – это объектно-ориентированный язык программирования, а данная парадигма программирования позволяет абстрактно описывать объекты, в том числе и математические объекты;
- .NET Core и .NET Standard – это современные, развивающиеся и востребованные кроссплатформенные технологии с открытым исходным кодом;
- Личные предпочтения автора.

Для поэтапного создания программы, были сформулированы и решены следующие задачи:

- Разработать библиотеку классов для объектов языка элементарной алгебры: символы алфавита, термы, формулы;
- Реализовать ввод логических формул;
- Разработать библиотеку классов для рациональных чисел и многочленов от одной вещественной переменной с рациональными коэффициентами;
- Реализовать алгоритм Тарского для базового случая;
- Реализовать алгоритм элиминации кванторов(ссылка на утверждение о сведении задачи к алгоритму тарского) для простых формул языка элементарной алгебры;
- Обеспечить минимальное покрытие юнит-тестами;
- Собрать все библиотеки и модули в единый программный комплекс.

Написание программы осуществлялось в среде разработки Microsoft Visual Studio 2019, доступной для использования в некоммерческих целях и скачивания на официальном сайте: https://visualstudio.microsoft.com/ru/.

Также была рассмотрена перспектива реализации алгоритма Тарского для общего случая.

2.1 Представление формул. Библиотека LogicLanguageLib

форма представления символов и формул языка

2.2 Система ввода. Класс Parser

Ввод формул

2.3 Реализация базового случая. Библиотека SimpleTarskiAlgorithmLib

Часть математики, часть алгоритм. Runner. Покрытие тестами. Консольное приложение

2.4 Библиотека SimpleTarskiAlgorithmRunner

SimpleTarskiAlgorithmRunner

2.5 Юнит-тестирование

тесты

2.6 Результаты работы

Демонстрация результатов

2.7 Общий случай

Часть математики. Описание нерешённых задач.

Заключение

Заключение

Список литературы

[1] Гибадулин Р. А. Алгоритм поиска вывода в Исчислении Высказываний и его программная реализация // Современные проблемы математики и информатики : сборник научных трудов молодых ученых, аспирантов и студентов. / Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль : ЯрГУ, 2019. – Вып. 19. – С. 28 – 37.

[2] Дурнев, В. Г. Элементы теории множеств и математической логики: учеб. пособие / Яросл. гос. ун-т. им. П. Г. Демидова, Ярославль, 2009 — 412 с.

[3] Матиясевич, Ю. В. Алгоритм Тарского // Компьютерные инструменты в образовании. – 2008. – № 6. – С. 14.

[4] Троелсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс; пер. с англ. и ред. Ю.Н. Артеменко. – 8-е изд. – М.; СПб.: Диалектика, 2020. – 1328 с.

[5] Якимова, О. П. Языки программирования. Ч.2: лабораторный практикум / О. П. Якимова, И. М. Якимов, В. Л. Дольников; Яросл. гос. ун-т им. П. Г. Демидова. – Ярославль : ЯрГУ, 2012. – 56 с.

[6] Tarski, A. A Decision Method for Elementary Algebra and Geometry: Prepared for Publication with the Assistance of J.C.C. McKinsey, Santa Monica, Calif.: RAND Corporation, R-109, 1951.

Добавить учебник по матану и алгебре

Приложение А

Приложение