

# **Математическая логика и теория алгоритмов**

# Цель

выяснить и проанализировать  
некоторые фундаментальные понятия,  
алгоритмы и методы, лежащие в основе  
информатики.

# Основные разделы

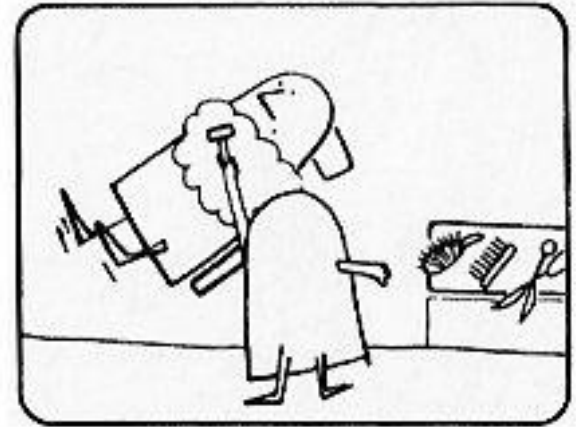
- математическая логика (главы 1 и 2),
- теория алгоритмов (главы 3, 4 и 5),
- теория сложности вычислений и эффективные алгоритмы (главы 6 и 7),
- **Учебное пособие:**  
Крючкова, Е.Н. Основы математической логики и теории алгоритмов: учебное пособие /Е. Н. Крючкова.- Барнаул : АлтГТУ , 2013 - 216 с. - Режим доступа:  
[http://new.elib.altstu.ru/eum/download/pm/Kruchkova\\_ml.pdf](http://new.elib.altstu.ru/eum/download/pm/Kruchkova_ml.pdf)

Рассмотрим несколько логических парадоксов, появление которых демонстрирует необходимость строго математического описания проблем и методов их решения.

# Парадокс парикмахера

В одном мужском монастыре живут только мужчины, и все мужчины этого монастыря бреют бороды. Некоторые умеют бриться сами, а некоторые — нет. Для этой цели в монастыре работает парикмахер, который также является членом этого монастыря. Парикмахер бреет тех и только тех людей, которые не бреются сами. Кто бреет

Парикмахер? Парикмахер бреет себя, то он бреется сам и не может себя брить по условиям работы парикмахера. Если он себя не бреет, то его должен брить парикмахер, то есть он сам должен брить себя.



# Парадокс лжеца

Вася говорит "Я лгу". Если Вася лжет, то сказанное не есть ложь, следовательно, он не лжет. Если Вася говорит правду, то сказанное им есть истина, следовательно, он лжет. В любом случае Вася лжет и не лжет одновременно.



# Парадокс Карри

Парадоксальный вывод из высказывания «Если это утверждение верно, то гоблины существуют». Вместо существования гоблинов может указываться любое неправдоподобное или ложное заявление (в английском оригинале — существование [Санта-Клауса](#)). Ход мыслей, ведущий к парадоксу, строится следующим образом:

Обозначим через  $S$  высказывание «Если  $S$  верно, то гоблины существуют»;

Мы не знаем, верно ли высказывание  $S$ . Но если бы высказывание  $S$  было верным, то это влекло бы существование гоблинов;

Но именно это и утверждается в высказывании  $S$ , таким образом  $S$  — верно;

Следовательно, гоблины существуют!



- **В строго формализованных теориях парадоксы такого типа не должны появляться!**
- Анализ парадоксов привел к выработке методов их устранения.
- Необоснованным является распространение некоторых свойств на все объекты без ограничения.
- **Мы тогда и только тогда можем согласиться с утверждением о существовании объекта, когда имеем алгоритм его построения.**



# Понятие алгоритма

- Слово "алгоритм" происходит от имени арабского математика Мохаммеда ибн Муса аль-Хорезми, который в IX столетии внес значительный вклад в разработку и практическое применение методов вычислений.

Все ли математические проблемы алгоритмически разрешимы?



# Необходимость математического определения алгоритма

1. Только при наличии формального определения алгоритма можно ставить задачу о разрешимости или неразрешимости каких-либо проблем. Если мы хотим доказать, что та или иная функция не является эффективно вычислимой, то мы прежде всего должны сформулировать точное математическое понятие эффективной вычислимости.
2. Необходимо иметь математически точный инструмент для сравнения различных алгоритмов решения одних и тех же задач, а также для сравнения различных проблем по сложности алгоритмов их решения. Такое сравнение невозможно без введения средств исследования алгоритмов как математических объектов и использования точного языка математики для их

# Что такое алгоритм?

- Прежде, чем ввести математически строгое определение алгоритма, необходимо рассмотреть свойства тех объектов, которые мы считаем алгоритмами.
- С точки зрения современной практики **алгоритм — это программа, а критерием алгоритмичности процесса является возможность его запрограммировать.**

# Свойства неформального понятия алгоритма

## 1. Конечность

Любой алгоритм задается последовательностью инструкций конечных размеров.

## 2. Детерминированность

Алгоритм выполняется детерминированно, т.е. представляющая алгоритм последовательность действий на одних и тех же данных всегда выполняется одинаково.

## 3. Дискретность

Отдельные инструкции алгоритма выполняются дискретно, т.е. последовательно по отдельным шагам, без использования каких-либо аналоговых устройств непрерывного принципа действия или соответствующих непрерывных методов.

# Свойства неформального понятия алгоритма

## 4. Вычислимость или эффективная вычислимость

Должен существовать вычислитель, способный выполнить указанные в алгоритме инструкции.

## 5. Конечная память

Вычислитель должен иметь средства для хранения и отображения информации. Память вычислителя может быть сколь угодно большой, но при каждом выполнении алгоритма требуется конечный объем памяти.

## 6. Результативность

Алгоритм должен быть результативным, т.е. завершающимся с некоторым результатом после некоторого конечного числа шагов.

# Неформальное понятие алгоритма

**Алгоритм** – это эффективная детерминированная конечная процедура, которую можно применять к любому элементу некоторого класса входов и которая для каждого такого входа дает в конце концов соответствующий результат.

# Типы алгоритмических моделей

- Можно выделить три основных типа универсальных алгоритмических моделей, различающихся исходными эвристическими соображениями относительно того, что такое алгоритм.
  1. Частично–рекурсивные функции
  2. Машины Тьюринга
  3. Алгоритмы Маркова

# Частично–рекурсивные функции

- Модель основана на функциональном подходе и рассматривает понятие алгоритма с точки зрения того, что можно вычислить с помощью алгоритма. Эта наиболее развитая и изученная модель является исторически первой формализацией понятия алгоритма и связывает определение алгоритма с наиболее традиционными понятиями математики — вычислениями и числовыми функциями.



# Машины Тьюринга

- Эта автоматная модель имеет в своей основе анализ процесса выполнения алгоритма и рассматривает алгоритм в виде набора инструкций для некоторой формальной модели вычислителя. Данный тип основан на представлении об алгоритме как о некотором детерминированном устройстве, способном выполнять в каждый отдельный момент лишь весьма примитивные операции. Такое представление не оставляет сомнений в однозначности алгоритма и элементарности его шагов. Кроме того, теоретическая основа этих моделей очень близка к ЭВМ.

# Алгоритмы Маркова

- Представляет собой языковой подход к понятию алгоритма, в основе которого лежит формализация процесса преобразования записей исходных данных в запись результатов.
- Всякие исходные данные для некоторой частной проблемы из какого-нибудь общего класса проблем могут быть сформулированы в виде некоторого выражения подходящего языка.
- Всякое решение этой проблемы в свою очередь также является выражением того или иного языка.
- Тогда алгоритм можно понимать как способ преобразования записи исходных данных в запись результатов.
- Преимущества такого типа формализации алгоритма — в его максимальной абстрактности и возможности применить понятие алгоритма к объектам произвольной природы, а не обязательно числовой, как это требуется для функциональной модели.

- На протяжении всего курса мы будем иметь дело с **неотрицательными целыми числами**, множествами неотрицательных целых чисел и отображениями неотрицательных целых чисел в неотрицательные целые числа. Если только не оговорено противное, мы используем термин "натуральное число" или просто "число" для обозначения неотрицательного целого числа.