

Элементы теории формальных языков

Лекция 1: От правил языка к компиляторам

Курс Б1.В.12

ИГУ, Кафедра информационных технологий

2025

Вы — будущие архитекторы мышления

Что общего?

- Правило «жи-ши»
- Проверка орфографии
- Поиск в Google
- Компиляция программы

Ваша роль как педагога

Не просто передать знания, а научить учеников **формально мыслить**

Кейс из школы: Кроссворд и шаблоны

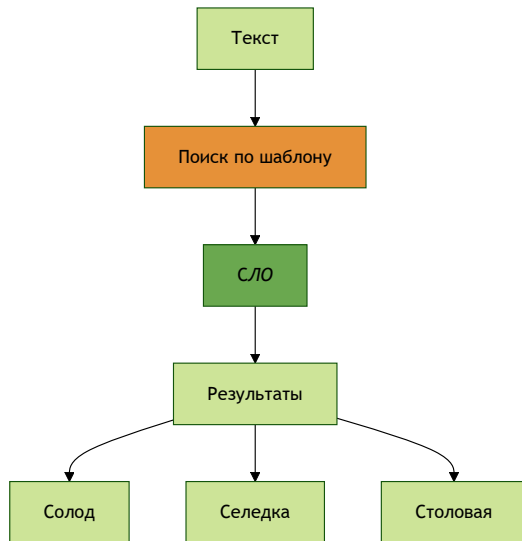
Задача

Найти слова по маске C^*LO^*

- Солод
- Селедка
- Столовая?

Связь с курсом

Это задача для **регулярных выражений** — тема 1-й лабораторной



Кейс из школы: Проверка выражений

Проблема

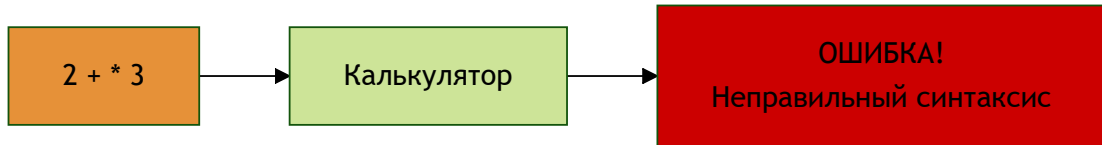
Почему калькулятор ругается на $2 + * 3$?

Нарушен синтаксис

Неправильная последовательность токенов

Связь с ЕГЭ

Задачи на анализ алгоритмов и формальное исполнение



Карта курса: Наш путь за семестр



Инструменты промышленного уровня

ANTLR4

- Генератор парсеров
- Используется в Twitter, Oracle, Hadoop

LLVM

- Платформа для компиляторов
- Используется в Clang, Rust, Swift

Что такое формальный язык?

Определение

Множество **строк** над некоторым **алфавитом**

Аналогия

- Алфавит: {А...Я, пробел, ...}
- Язык: Все грамматически правильные предложения

Ключевой вывод

Язык — это не набор символов, а набор **правильно построенных фраз**

Что такое формальный язык?

Формальное определение

Формальный язык — это:

- Множество **строк** (слов)
- Над конечным **алфавитом**
- Заданное **грамматикой**

Компоненты грамматики

- V — алфавит (терминалы)
- N — нетерминалы
- P — правила вывода
- S — начальный символ

Пример грамматики арифметики

- $V = \{+, *, (,), \text{id}\}$
- $N = \{E\}$
- $S = E$
- $P : E \rightarrow E + E$
- $E \rightarrow E * E$
- $E \rightarrow (E)$
- $E \rightarrow \text{id}$

Вывод в формальных грамматиках

Левый вывод (Leftmost)

- Всегда заменяем **самый левый** нетерминал
- **Последовательность:**
 - 1 E
 - 2 $E + E$
 - 3 $\text{id} + E$
 - 4 $\text{id} + E * E$
 - 5 $\text{id} + \text{id} * E$
 - 6 $\text{id} + \text{id} * \text{id}$

Правый вывод (Rightmost)

- Всегда заменяем **самый правый** нетерминал
- **Последовательность:**
 - 1 E
 - 2 $E * E$
 - 3 $E * \text{id}$
 - 4 $E + E * \text{id}$
 - 5 $E + \text{id} * \text{id}$
 - 6 $\text{id} + \text{id} * \text{id}$

Важно для педагогов!

Левый вывод = нисходящий разбор | Правый вывод = восходящий разбор

Практическое применение: зачем это учителю?

В школьной информатике

- **Объяснение синтаксиса:** Почему $2 + *3$ — ошибка?
- **Построение выражений:** Как калькулятор понимает приоритет операций?
- **Разбор алгоритмов:** Задачи ЕГЭ на формальное исполнение

В университетском курсе

- **LL-анализаторы:** Используют левый вывод
- **LR-анализаторы:** Используют правый вывод
- **ANTLR4:** Генерирует LL()-парсеры
- **Bison/Yacc:** Генерируют LR-парсеры

Живой пример для урока

Задание: Постройте левый и правый вывод для выражения: $(a + b) * c$

Формальная грамматика — правила игры

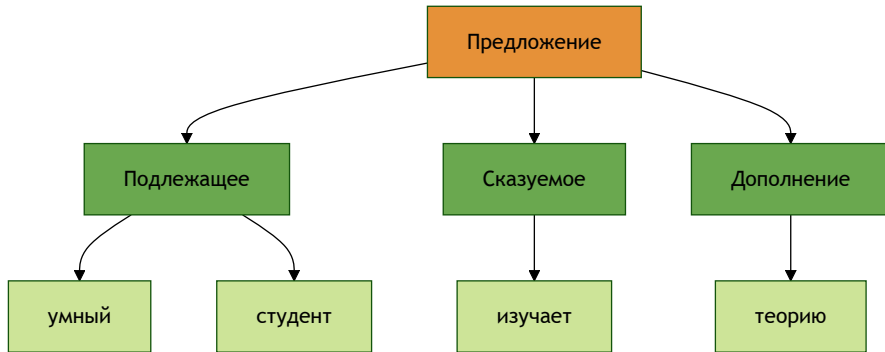
Русский язык

- ПРЕДЛОЖЕНИЕ \rightarrow ПОДЛЕЖ СКАЗУЕМ
- ПОДЛЕЖ \rightarrow МЕСТОИМ | СУЩ
- МЕСТОИМ \rightarrow "я" | "ты" | "он"

Арифметика

- ВЫРАЖЕНИЕ \rightarrow ЧИСЛО | ВЫР
ОПЕРАТОР ВЫР
- ОПЕРАТОР \rightarrow "+" | "-" | "*" | "/"

Живой пример: Разбор предложения



Вывод

Точно так же парсер разбирает $x = 2 + 3 * 5$ на дерево

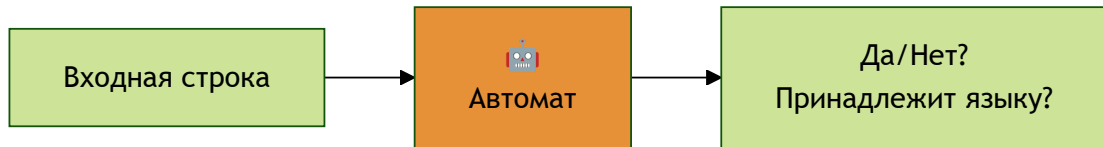
Что такое автомат?

Определение

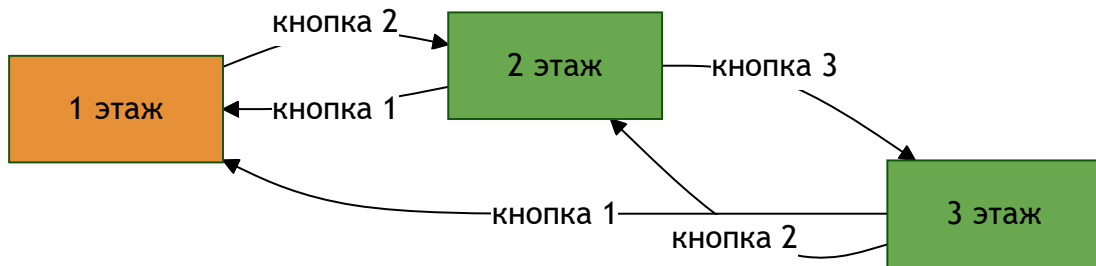
Устройство с конечной памятью, читает строку посимвольно

Ключевая аналогия

Робот-проверяющий — принимает решение, принадлежит ли строка языку



Кейс: Лифт как конечный автомат



Применение

Игры, пользовательские интерфейсы, протоколы связи

Кейс: Распознаватель чисел

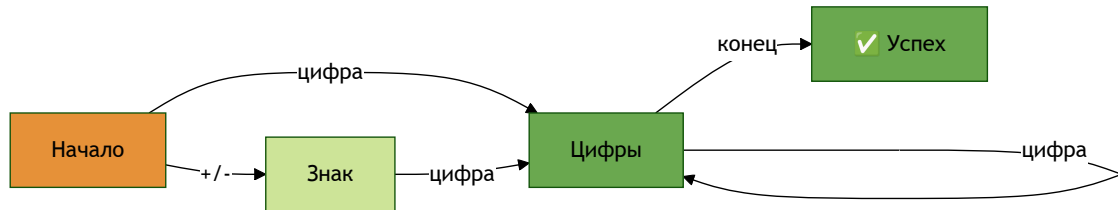
Задача

Определить, является ли строка целым числом

- "123" ☒
- "-45" ☒
- "12a3" ☒

Связь с курсом

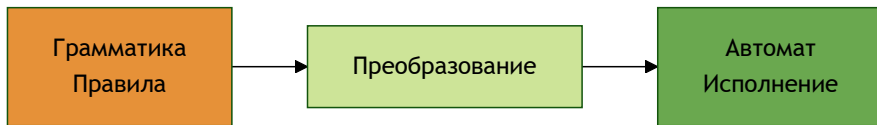
Это и есть **лексический анализатор** для чисел!



Связка: Грамматика \rightarrow Автомат

Фундаментальный принцип

Для каждого класса грамматик существует свой класс автоматов



Иерархия Хомского — классификация формальных языков

Тип 0: Неограниченные

- **Автомат:** Машина Тьюринга
- **Примеры:** Все языки программирования, искусственный интеллект
- **Ограничения:** Нет ограничений

Тип 2: КС-грамматики

- **Автомат:** МП-автомат
- **Примеры:** Синтаксис Python, Java, C++
- **Ограничения:** Контекстно-свободные правила

Тип 1: Контекстно-зависимые

- **Автомат:** Линейно ограниченный автомат
- **Примеры:** Естественные языки, анализ кода
- **Ограничения:** Зависимость от контекста

Тип 3: Регулярные

- **Автомат:** Конечный автомат
- **Примеры:** Регулярные выражения, лексический анализ
- **Ограничения:** Самые строгие

Фундаментальный принцип

Каждый класс грамматик соответствует своему классу автоматов

Иерархия Хомского в реальном мире

Регулярные (Type 3)

- Регулярные выражения
- Лексический анализ

Контекстно-зависимые (Type 1)

- Естественные языки
- Анализ кода

КС-грамматики (Type 2)

- Синтаксис Python, Java
- МП-автоматы

Неограниченные (Type 0)

- Искусственный интеллект
- Системы принятия решений

Фокус: Регулярные языки

Их автомат

Конечный автомат (ДКА/НКА)

Их грамматика

Очень ограниченная

Их инструмент

Регулярные выражения — невероятно полезны на практике

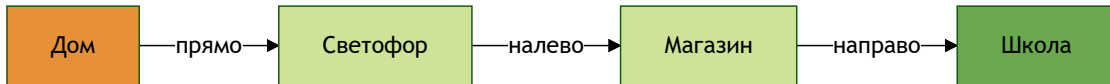
ДКА — робот с одним путем

Принцип

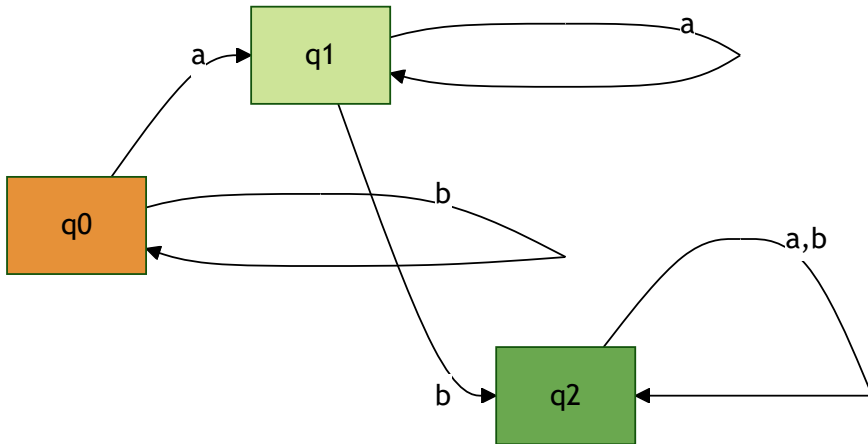
Для каждого состояния и символа — **ровно один** переход

Аналогия

Маршрут такси с четкими инструкциями



Пример ДКА: Поиск окончания 'ab'



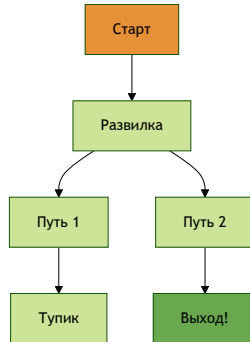
НКА — робот с воображением

Принцип

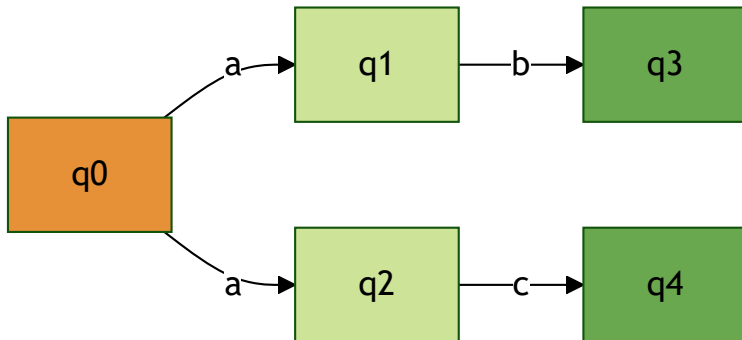
Может угадывать, несколько вариантов перехода

Аналогия

Лабиринт с "сохранениями" — пробуем разные пути



Пример НКА: 'ab' или 'ac'



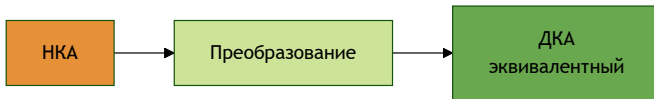
Принцип

Если хоть одна ветка ведет к успеху — строка принята

Теорема: НКА \rightarrow ДКА

Фундаментальный факт

Любой НКА можно преобразовать в эквивалентный ДКА



Вывод

Мощность у них **одинаковая**

Регулярные выражения — язык шаблонов

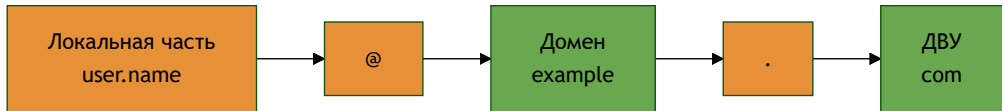
Базовые операции

- Конкатенация: ab
- Выбор: $a|b$
- Итерация: a^*

Примеры

- a^* — любое количество 'а'
- $(a|b)^*$ — строка из 'а' и 'b'
- $a+b+$ — 'а' затем 'b'

Кейс: Валидация email (упрощенная)



Применение

Формы регистрации, проверка вводимых данных

Кейс: Извлечение хештегов

Задача

Извлечь все хештеги из поста

Это лексический анализ!

То, что мы делаем в Лабораторной 1

Шаблон Python

```
r"#\w+"
```

- #формальные_языки ☒
- #информатика ☒
- #ИГУ ☒

Минимизация ДКА — оптимизация

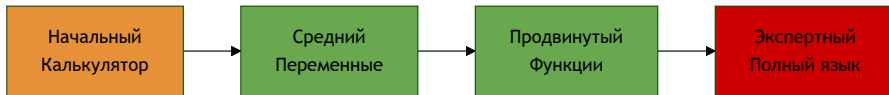
Зачем?

- Эффективность
- Меньше состояний
- Быстрее работа

Педагогическая аналогия

Оптимизация плана урока — убрать лишнее, оставить суть

Лабораторный практикум: Сквозной проект



Уникальная возможность

17 вариантов сложности — от калькулятора до своего языка

Лабораторная 1: Создаем лексер

Цель

Написать регулярные выражения для разбивки кода на токены

Инструмент

ANTLR4 — генератор парсеров

Вход/Выход

- Вход: Текст на мини-языке
- Выход: Поток токенов

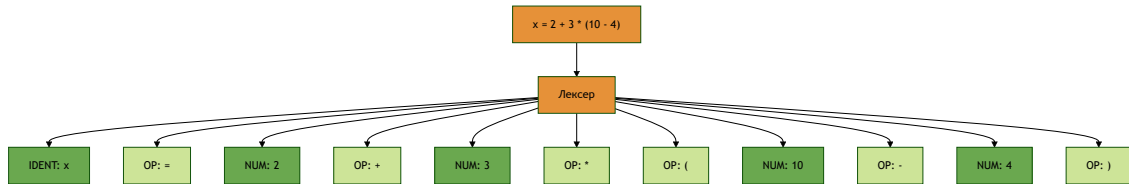
Пример кейса: Калькулятор

Исходный код

$x = 2 + 3 * (10 - 4)$

Токены лексера

- IDENT(x)
- OP(=)
- NUM(2)
- OP(+)
- ...



Грамматика для лексера в ANTLR4

Фрагмент грамматики

```
NUMBER : [0-9]+ ;  
IDENTIFIER : [a-zA-Z]+ ;  
PLUS : '+' ;  
ASSIGN : '=' ;  
WS : [ \t\r\n]+ -> skip ;
```

Объяснение

Мы просто дали имена нашим регулярным выражениям — это не страшно!

Основные инструменты

- Git — контроль версий
- Docker — универсальная среда
- Pytest — тестирование

AI (DeepSeek)

- Контролируемое использование!
- Генерация тестов
- Объяснение ошибок
- Не для написания кода

Педагогический бонус: Идея для школы

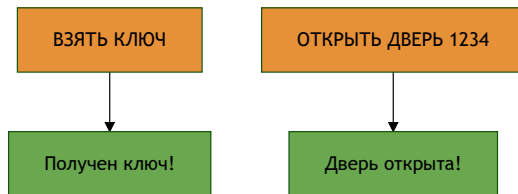
Текстовый квест

Создайте "язык" для описания квестов:

- ВЗЯТЬ КЛЮЧ
- ОТКРЫТЬ ДВЕРЬ 1234

Результат

Ученики видят прямое применение абстрактной теории в игровой форме



Что дальше? Анонс Лекции 2

Проблема

Регулярных выражений НЕ ХВАТАЕТ для:

- Проверки скобок ((()))
- Конструкций if...then...else

Решение

Контекстно-свободные грамматики и
МП-автоматы

Тема лекции 2

«Синтаксис: от предложений до AST»

Ключевые выводы

1. Формальные языки — основа IT-индустрии
2. Связка «Грамматика → Автомат» фундаментальна
3. Регулярные выражения — мощный инструмент

4. Курс — путь к созданию компилятора
5. Вы сможете перенести знания в школьные проекты

Обсудим

- Что самое сложное?
- Идеи для школьных проектов?
- Какой кейс выбрали бы вы?

Мотивирующая цитата

«Чтобы овладеть языком программирования, нужно его выучить. Чтобы понять саму природу языков, нужно научиться их создавать.»