# Теория автоматов и формальных языков

## Магазинные автоматы

**Автор:** Екатерина Вербицкая

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

21 ноября 2019

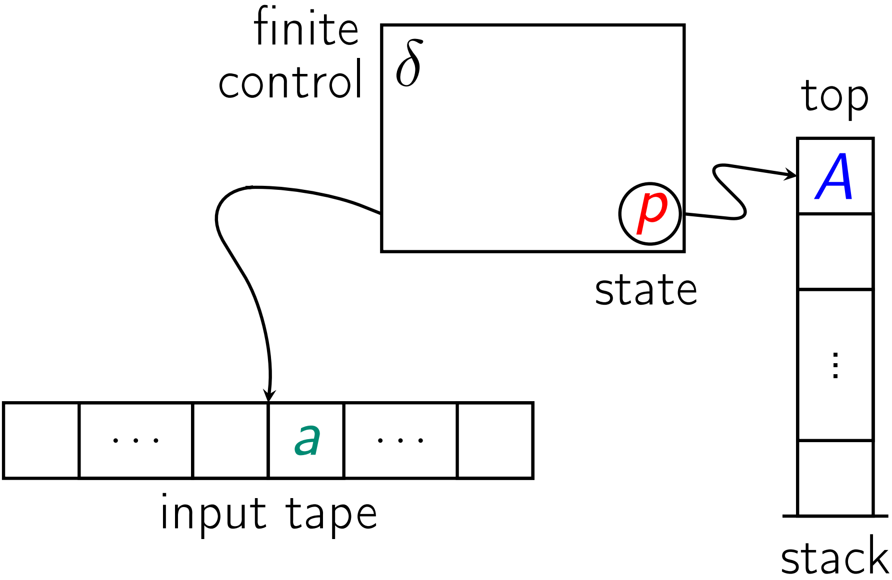
# В предыдущей серии

- Регулярные языки распознаются с помощью конечных автоматов
- Разные алгоритмы синтаксического анализа для контекстно-свободных языков
  - CYK
  - Рекурсивный спуск
  - LL(1)
  - LR(0), SLR(1), CLR(1), LALR(1)
- Есть ли универсальный распознаватель для КС-языков?

# TLDR

- Произвольный КС язык можно распознать при помощи магазинного автомата (он же автомат с магазинной памятью, он же pushdown automata, он же pda)
- Магазинный автомат по сути — автомат со стеком
- Детерминированные магазинные автоматы могут распознавать только детерминированные КС языки
- Недетерминированные магазинные автоматы могут распознавать произольные КС языки

# Что такое магазинный автомат



# Что такое магазинный автомат: неформально

- Автомат, переходы которого осуществляются по входному символу, текущему состоянию и символу на вершине стека
  - У конечного автомата не было стека
- Никакие состояния стека, кроме вершины, не доступны
- Во время перехода может изменяться стек
  - Положить что-то на стек (push)
  - Снять верхушку со стека (pop)
- А может и не изменяться
  - Магазинный автомат может вообще игнорировать стек
  - Или стек может не изменяться, хоть значение оттуда и читается
- Итого: по тройке (входной символ, состояние, символ на вершине стека) получается новое состояние, и модифицируется (или нет) стек

# Формальное определение

*Недетерминированный* магазинный автомат это $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$

- $Q$ — конечное множество состояний
- $\Sigma$ — конечное множество символов, входной алфавит
- $\Gamma$ — конечное множество символов, стековый алфавит
- $\delta : Q \times (\Sigma \cup \varepsilon) \times \Gamma \to 2^{Q \times \Gamma^*}$ — функция переходов
- $q_0 \in Q$ — стартовое состояние
- $Z \in \Gamma$ — начальный элемент стека
- $F \subseteq Q$ — множество принимающих (конечных) состояний

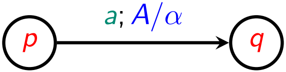
# Отношение переходов

$$(q, \alpha) \in \delta(p, a, A) \text{ означает}$$

- Если магазинный автомат находится в состоянии $p \in Q$,
- на вершине стека находится $A \in \Gamma$,
- а со входа читается символ $a \in \Sigma \cup \varepsilon$,

- то изменяем состояние на $q \in Q$,
- снимаем со стека символ $A$, записываем на стек строку $\alpha \in \Gamma^*$

- $\Sigma \cup \varepsilon$ сигнализирует о том, что вход можно и не читать



# Семантика магазинного автомата

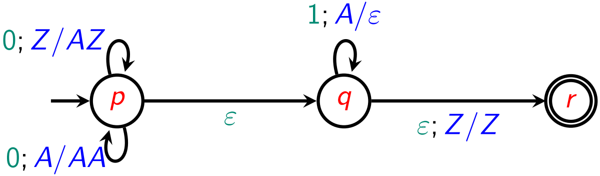
- Мгновенное описание МА: $(p, \omega, \beta) \in Q \times \Sigma^* \times \Gamma^*$
  - $p$ — текущее состояние автомата
  - $\omega$ — непрочитанный фрагмент входного потока
  - $\beta$ — содержимое стека (верхушка записана первой)
- Отношение $\vdash$ на мгновенных описаниях (шаг)
  - Для каждого $(q, a) \in \delta(p, a, A)$, верно $(p, ax, A\gamma)) \vdash (q, x, \alpha\gamma))$ для произвольных $x \in \Sigma^*, \gamma \in \Gamma^*$
- Шаг не определен, если стек пуст

# Семантика магазинного автомата: вычисление

- Вычисление — последовательность шагов
- Начальное мгновенное описание $(q_0, \omega, Z)$
- Выбирается любой из подходящих шагов
- Если какой-нибудь выбор приведет к успеху, значит, строка распознается

- Два варианта окончания работы
  - По достижении конечного состояния
    - $L(M) = \{\omega \in \Sigma^* \mid (q_0, \omega, Z) \vdash^* (f, \varepsilon, \gamma), f \in F, \gamma \in \Gamma^*\}$
  - По опустошении стека
    - $N(M) = \{\omega \in \Sigma^* \mid (q_0, \omega, Z)) \vdash^* (q, \varepsilon, \varepsilon), q \in Q\}$
  - Эти варианты эквивалентны: по автомату, завершающемуся по первой схеме, можно посмотроить автомат, завершающийся по второй схеме, и наоборот
- $\vdash^*$ — транзитивно рефлексивное замыкание отношения $\vdash$

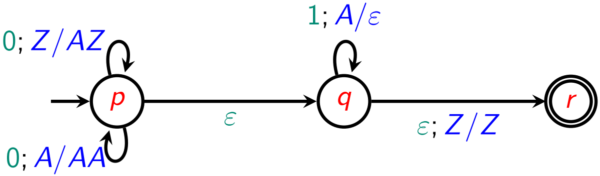
# Пример: язык $\{0^n 1^n \mid n \geq 0\}$




Вычисление на строке 0011:

- $(p, 0011, Z) \vdash (q, 0011, Z) \vdash (r, 0011, Z)$ — провал
- $(p, 0011, Z) \vdash (p, 011, AZ) \vdash (q, 011, AZ)$ — провал
- $(p, 0011, Z) \vdash (p, 011, AZ) \vdash (p, 11, AAZ) \vdash (q, 11, AAZ) \vdash (q, 1, AZ) \vdash (q, \varepsilon, Z) \vdash (r, \varepsilon, Z)$ — успех (по принимающему состоянию)

# Пример: язык $\{0^n 1^n \mid n \geq 0\}$




Вычисление на строке 00111:

- $(p, 00111, Z) \vdash (q, 00111, Z) \vdash (r, 00111, Z)$ — провал
- $(p, 00111, Z) \vdash (p, 0111, AZ) \vdash (q, 0111, AZ)$ — провал
- $(p, 00111, Z) \vdash (p, 0111, AZ) \vdash (p, 111, AAZ) \vdash (q, 111, AAZ) \vdash (q, 11, AZ) \vdash (q, 1, Z) \vdash (r, 1, Z)$ — провал

# Формальное определение ДМА

*Детерминированный* магазинный автомат это $(Q, \Sigma, \Gamma, \delta, q_0, Z, F)$

- $Q$ — конечное множество состояний
- $\Sigma$ — конечное множество символов, входной алфавит
- $\Gamma$ — конечное множество символов, стековый алфавит
- $\delta : Q \times (\Sigma \cup \varepsilon) \times \Gamma \to Q \times \Gamma^*$ — функция переходов
  - $\forall q \in Q, Z \in \Gamma$ : если есть эпсилон-переход $\delta(q, \varepsilon, Z)$ то нет переходов по терминалам $\delta(q, a, Z)$
- $q_0 \in Q$ — стартовое состояние
- $Z \in \Gamma$ — начальный элемент стека
- $F \subseteq Q$ — множество принимающих (конечных) состояний

# Детерминированные магазинные автоматы vs недетерминированные

- В общем случае одной входной строке может соответствовать несколько вычислений
  - Некоторые из них могут завершаться в принимающих состояниях
- Если существует хотя бы одно вычисление, завершающееся в принимающем состоянии, строка принадлежит языку
- Если для каждой строки существует ровно одно вычисление в магазинном автомате, то он является *детерминированным*
  - Соответствующий язык является *детерминированным КС языком*
- Детерминированный магазинный автомат является частным случаем недетерминированного, поэтому детерминированные КС языки — строгое подмножество контекстно-свободных

# Неэквивалентность двух видов приема для детерминированных магазинных автоматов

*Беспрефиксный язык* — язык, в котором никакое слово не является префиксом другого

- Прием языка детерминированным магазинным автоматом по пустому стеку и по допускающему состоянию эквивалентно только для беспрефиксных языков
- Рассмотрим слово $\omega = \alpha\beta : \alpha, \beta \in \Sigma^*, \omega, \alpha \in L$, где $L \subseteq \Sigma^*$
- При попытке распознать слово $\omega$ ДМП завершит свою работу, как только прочитает $\alpha$
- $\omega$ никогда не будет принята
- Можно построить ДМП, принимающий по допускающему состоянию, который допускает префиксный язык

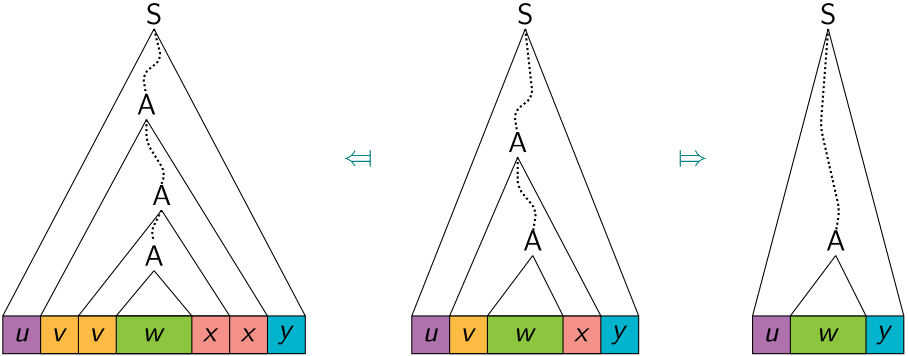
# Построение магазинного автомата по КС-грамматике

- Интуиция:
  - Для каждого нетерминала: заменяем его на стеке на правую часть правила
  - Для каждого терминала: считываем со входа этот терминал и кладем его на стек
- Построение:
  - Для каждого правила $A \to \alpha$ добавляем $(1, \alpha)$ в $\delta(1, \varepsilon, A)$
  - Для каждого терминала $a$ добавляем $(1, \varepsilon)$ в $\delta(1, a, a)$
- Относительно бесполезный автомат: как найти правильное вычисление?

# Лемма о накачке для КС языков

**Теорема**

*Если язык $L$ является контекстно-свободным, то*
$\exists p \geq 1 : \forall s \in L : |s| \geq p$ *можно разбить на подстроки*
$s = uvwxy : |vwx| \leq p, |vx| \geq 1$ *и*
$\forall n \geq 0.\, uv^n wx^n y \in L$



# Лемма о накачке для КС языков: пример

Язык $L = \{a^n b^n c^n\}$
Предполагаем, что он КС, тогда по Лемме существует $p$...
Рассмотрим слово $a^p b^p c^p = uvwxy, |vwx| \leq p, |vx| \geq 1$

- $vwx = a^j, j \leq p$
- $vwx = a^j b^k, j + k \leq p$
- $vwx = b^j, j \leq p$
- $vwx = b^j c^k, j + k \leq p$
- $vwx = c^j, j \leq p$

Строка $uv^i wx^i y$ не содержит одинаковое количество букв для всех $i$.
Например, рассмотреть $i = 2$. Получили противоречие — успех