

# Задание 1

## Регулярные языки и автоматы

### Литература:

1. Хопкрофт Д., Мотвани Р., Ульман Д.  
Введение в теорию автоматов, языков и вычислений.  
М.: Вильямс, 2002.
2. Ахо А., Ульман Д.  
Теория синтаксического анализа, перевода и компиляции  
М.: Мир, 1978. Гл. 0, 2.
3. Серебряков В.А., Галочкин М.П., Гончар Д.Р., Фуругян М.Г.  
Теория и реализация языков программирования.  
М.: МЗ-пресс, 2006.

**Ключевые слова**<sup>1</sup>: принцип мат. индукции, язык, регулярные выражения, конкатенация, объединение, итерация, конечные автоматы (КА), детерминированные и недетерминированные КА, регулярные языки.

## 1 Т<sub>E</sub>X

В этом году все задания принимаются только в формате Т<sub>E</sub>X. Во-первых, через пару лет вам уже предстоит написание диплома и большинство дипломов так или иначе связаны с математикой, и уж точно в подавляющем большинстве из них присутствуют формулы. Т<sub>E</sub>X – довольно гибкий инструмент для работы с математическими текстами, он является стандартом для публикаций в крупных журналах, не зависит от платформы и с ним довольно удобно работать. Даже в переписках в сети, уже принято записывать математические формулы в стиле те<sub>X</sub> и большинство профильных сайтов поддерживают конвертацию на лету из те<sub>X</sub> в формулы. Я буду предоставлять те<sub>X</sub>овские исходники ко всем заданиям. Во-вторых, в этом году я планирую частично автоматизировать процесс приёма задания, поэтому, если планы сбудутся, то с

---

<sup>1</sup>минимальный необходимый объём понятий и навыков по этому разделу)

определённого момента сдавать задания в техе придётся по техническим причинам.

В качестве литературы по теку, я рекомендую [книгу](#) Львовского и [статью с набором примеров](#) Воронцова и [wiki-учебник](#). Так же я нашёл [видеоуроки](#), возможно они окажутся полезны.

Техом пользуется столько людей, что практически все потребности, которые возникают в процессе написания научных текстов, уже удовлетворены. Например, для построения графов автоматов есть множество пакетов. Я предпочитаю пакет *tikz*. Документация к этому пакету находится [здесь](#).

В качестве редактора, я рекомендую использовать [Texmaker](#). Это довольно удобный кроссплатформенный редактор.

## 2 О заданиях

Задания сдаются еженедельно, сдать задание надо до 22:59 среды. Исходники тека и pdf с заданием нужно присылать по адресу [homework@rubtsov.su](mailto:homework@rubtsov.su), если на сайте не появятся иные указания, для писем по другому поводу стоит использовать адрес [alex@rubtsov.su](mailto:alex@rubtsov.su). В самом задании обязательно **укажите фамилию, имя и номер группы**.

Задачи делятся на следующие категории: предбазовые, базовые, усиленные и дополнительные. Если вы не можете решить задачи предбазового уровня, то скорее всего вам стоит ещё раз разобраться с материалом. Сдавать эти задачи крайне желательно, благо они не большие по объёму, чтобы в случае критической ошибки я мог дать обратную связь. Задачи базового уровня – это обязательные задачи. Они не представляют собой особой трудности, но их уже необходимо сдавать в обязательном порядке. Усиленные задачи – это задачи в рамках курса и близкие к курсу, которые крайне желательно решать, если вы хотите автомат: успешное решение этих задач облегчит получение автомата и тем более сдачу задания.

Каждую неделю будут проводиться контрольные на 10-15 минут. Формат – оренбук: вы можете пользоваться любой литературой, но *исключительно в бумажном виде!* Оценки за контрольные 0 и от 3 до 7. Без дополнительного общения, я готов в качестве автомата выставить среднюю оценку за контрольные, если у вас нет штрафных очков. Но обычно, те кто не имеют штрафных очков желают повысить оценку, поэтому среднюю оценку по контрольным следует рассматривать как оценку с

которой начинается разговор. Автомат можно получить только если промежуточные общие контрольные для всего курса написаны не на два.

За несданные задания и плохо написанные контрольные начисляются штрафные очки, которые на очных сдачах преобразуются в дополнительные задачи.

Не сданное в срок задание или не написанную в срок семинарскую контрольную нельзя пересдать потом, вне зависимости от причины. Все набранные долги можно будет закрыть на очной сдаче, поэтому дополнительные меры для увеличения мировой справедливости применяться не будут.

### 3 Теоретико-множественное отступление

Для полноценного прохождения данного курса, вы должны помнить азы теории множеств. Вы должны знать что такое множество, пустое множество, элемент множества, основные операции над множествами: объединение, пересечение, дополнение, исключение. Однако, как показывает практика, обязательно найдутся люди, которые не помнят что такое декартово (прямое) произведение.

**Определение 1.** Декартовым произведением двух множеств  $X$  и  $Y$  называют множество  $X \times Y = \{(x, y) \mid x \in X, y \in Y\}$ .

Также нередко забывают, что множество всех подмножеств множества  $X$ , обозначают  $2^X$ .

Восполнить пробелы в теории множеств и дискретном анализе, а также узнать много нового и интересного можно из книги Н.К. Верещагина и А. Шеня *Лекции по математической логике и теории алгоритмов*. Эта книга, а также много других интересных и полезных книг, находятся в свободном доступе по адресу <http://www.mccme.ru/free-books/>.

### 4 Регулярные языки и конечные автоматы

Под *алфавитом* понимается конечное множество. Мы будем обозначать алфавиты заглавными греческими буквами, такими как  $\Sigma, \Gamma$ . Обычно мы будем использовать двухбуквенные алфавиты  $\Sigma = \{a, b\}$  или  $\Sigma = \{0, 1\}$ . Множества мы будем обозначать заглавными буквами, а их элементы строчными.

Для букв – элементов алфавита – определена операция *конкатенации*:  $a \cdot b = ab$ . Слово  $w = w_1w_2 \dots w_n$  – конкатенация букв. Длину слова  $w$ , мы будем обозначать  $|w|$ . Для слова  $w = w_1w_2 \dots w_n$ , длина  $|w|$  равна  $n$ ,  $i$ -ый символ слова мы будем обозначать  $w[i] = w_i$ , подслово  $w_iw_{i+1} \dots w_j$  будем обозначать  $w[i, j]$ . Мы не будем различать слова длины 1 и буквы, а также одноэлементные множества слов и слова. Пустое слово мы будем обозначать  $\varepsilon$ . Пустое слово обладает следующими свойствами: Для любого слова  $w$ ,  $\varepsilon \cdot w = w \cdot \varepsilon = w$ ,  $|\varepsilon| = 0$ . Для множеств слов определены следующие операции:

- Конкатенация:  $X \cdot Y = \{x \cdot y \mid x \in X, y \in Y\}$ .
- Возведение в степень:  $X^n = \underbrace{X \cdot X \cdot \dots \cdot X}_n$
- Объединение:  $X|Y = X + Y = X \cup Y = \{w \mid w \in X \text{ или } w \in Y\}$ .
- Итерация<sup>2</sup>  $X^* = \varepsilon + X + X^2 + X^3 + \dots + X^n + \dots$

В первой части курса мы будем изучать класс регулярных языков REG. Регулярные языки определяются следующим образом:

- $\emptyset \in \text{REG}$ .
- $\forall \sigma \in \Sigma : \{\sigma\} \in \text{REG}$ .
- $\forall X, Y \in \text{REG} : X \cdot Y, X|Y, X^* \in \text{REG}$ .
- Больше нет регулярных языков.

Так, множество всех слов над алфавитом  $\Sigma$  обозначают  $\Sigma^*$ .

Запись регулярных языков с использованием скобок и определённых выше операций, называют *регулярным выражением*. Сами по себе скобки являются разделителем и не несут смысловой нагрузки: Регулярное выражение  $(a)$  задаёт язык  $\{a\}$ ,  $(a|b) = \{a, b\}$ ,  $(a|b)^* = \{a, b\}^* = \Sigma^*$ .

**Упражнение 1.** Показать, что  $\varepsilon \in \text{REG}$ .

В теории формальных языков, каждому классу языков соответствует модель вычислений, которая распознаёт данный класс языков. Для класса регулярных языков такой моделью является *Конечный Автомат (КА)*.

---

<sup>2</sup>Эта операция также носит название звёздочка Клини, в честь выдающегося математика Стивена Клини.

**Определение 2.** Конечный автомат  $\mathcal{A}$  – это устройство, описываемое набором  $(Q, \Sigma, q_0, \delta, F)$ , где

- $Q$  – конечное множество состояний автомата;
- $\Sigma$  – алфавит, слова над которым обрабатывает автомат;
- $q_0$  – начальное состояние автомата;
- $\delta : Q \times \Sigma \rightarrow 2^Q$  – функция переходов;
- $F \subset Q$  – множество принимающих состояний.

Автомат является *детерминированным* (ДКА), если функция переходов определена однозначно, то есть для каждого состояния  $q$  и для каждого символа  $\sigma$ , существует не больше одного состояния  $q' \in \delta(q, \sigma)$ . В противном случае, автомат является *недетерминированным* (НКА).

На вход автомата подаётся слово  $w = w_1 \dots w_n$ . Автомат обрабатывает слово слева направо по тактам.

В детерминированном случае, за такт работы автомат находится в состоянии  $q$ , считывает символ  $\sigma$  и вычисляет функцию перехода  $\delta(q, \sigma)$ . Если  $\delta(q, \sigma) = q'$ , то автомат переходит в состояние  $q'$ , если же  $\delta(q, \sigma) = \emptyset$ , то автомат прекращает работу.

В недетерминированном случае, за такт работы, автомат *недетерминированно выбирает состояние*  $q'$  из множества  $\delta(q, \sigma)$  и переходит в состояние  $q'$ . В случае, если  $\delta(q, \sigma) = \emptyset$ , автомат прекращает работу. Под недетерминированным выбором, мы понимаем, следующую ситуацию.

Слово *принимается автоматом*, если после обработки слова, автомат оказался в принимающем состоянии. Недетерминированный автомат *всегда* оказывается в принимающем, состоянии, если он может в него попасть. Таким образом, детерминированный автомат принимает слово, если после обработки слова он оказался в принимающем состоянии, а недетерминированный автомат принимает слово, если существует такая последовательность выборов состояний, что после обработки слова он оказывается в принимающем состоянии.

Назовём *конфигурацией* пару  $(q, w) \in Q \times \Sigma^*$ . На множестве конфигураций введём соответствующее тактам работы автомата бинарное отношение  $\vdash$ : для всех  $q' \in \delta(q, a)$  и для всех  $w \in \Sigma^*$   $(q, aw) \vdash (q', w)$ . Рефлексивное и транзитивное замыкание отношения  $\vdash$  обозначим  $\vdash^*$ . Таким образом, автомат принимает слово, если существует такая цепочка

конфигураций

$$(q_0, w_1w[2, n]) \vdash (q'_1, w[2, n]), (q'_1, w_2w[3, n]) \vdash (q'_2, w[3, n]), \dots, (q'_{n-1}, w_n) \vdash (q'_n, \varepsilon),$$

что  $q'_n \in F$ . Или в терминах транзитивного замыкания  $\exists q'_n : (q_0, w) \vdash^* (q'_n, \varepsilon)$ .

Множество всех слов, принимаемых автоматом  $\mathcal{A}$  будем обозначать  $L(\mathcal{A})$ . Автомат  $\mathcal{A}$  *принимает* язык  $L$ , если  $\forall w \in L : w \in L(\mathcal{A})$ , другими словами  $L \subseteq L(\mathcal{A})$ . Если при этом,  $L(\mathcal{A}) \subseteq L$ , то будем говорить, что автомат  $\mathcal{A}$  *распознаёт* язык  $L$ . Здесь есть тонкая лингвистическая разница, слова “принимает” и “распознаёт”, конечно схожи, но за ними стоят разные определения.

## 5 О решении задач

Я хочу отметить, что под *решением задачи* понимается не получение конечного ответа без пояснений, а последовательность *обоснованных* действий, приводящих к правильному ответу. Все действия и построения должны быть обоснованы *всегда*. Часто я буду акцентировать в условии внимание на том, что именно нужно доказать, но это не означает, что если в задаче нет фразы в духе «доказать, что  $A = B$ », то можно писать не подкреплённый доводами поток сознания.

Не обязательно решить все задачи, поэтому если не получается что-то решить самостоятельно, не надо это судорожно списывать. Лучше решить то, что получается. Не страшно не решить – страшно попасться на списывании. Я буду кластеризовать работы по глупостям. Это не означает, что задачи нельзя обсуждать друг с другом. Если кто-то рассказал вам идею решения и вы думаете, что её поняли и решили записать, то укажите ссылку на автора идеи.

## 6 Задачи предбазового уровня

**Задача 1.**

1.  $\{a, aa\} \cdot \{b, bb\} = ?$
2.  $\{a, aa\} + \{b, bb\} = ?$
3.  $\{a, aa\} \times \{b, bb\} = ?$

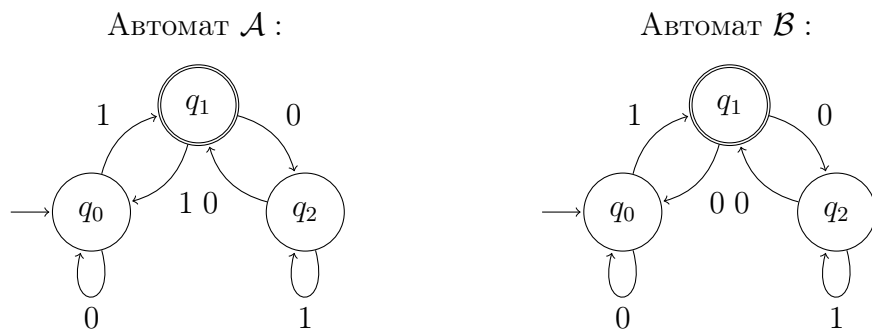
4.  $((aa|b)^*(a|bb)^*)^* = ?$
5.  $\{a^{3n} | n > 0\} \cap \{a^{5n+1} | n \geq 0\}^* = ?$
6.  $\emptyset \cap \{\varepsilon\} = ?$

## 7 Задачи базового уровня

Если в задаче явно не указан алфавит, но в тексте упомянуты символы  $a, b$  ( $0, 1$ ), то алфавит в этой задаче состоит из символов  $a, b$  ( $0, 1$ ).

**Задача 2.** Записать регулярное выражение для языка  $L = \Sigma^* \setminus \{(a|b)^* ab(a|b)^*\}$ . Доказать, что язык порождённый регулярным выражением совпадает с  $L$ .

**Задача 3.**



Для каждого автомата ответьте на следующие вопросы (1-2):

1. Автомат задан через граф переходов. Запишите определение автомата в виде  $(Q, \Sigma, \delta, q_0, F)$ . Опишите элементы каждого множества
2. Является ли автомат детерминированным?
3. Опишите последовательность конфигураций автомата  $\mathcal{A}$  при обработке слова  $w = 1101010$ . Верно ли, что  $w \in L(\mathcal{A})$ ?
4. Принимает ли автомат  $\mathcal{B}$  слово  $v = 10001$ ?
5. Укажите по одному слову, принадлежащему  $L(\mathcal{A})$ ,  $L(\mathcal{B})$  и по одному слову, не принадлежащему  $L(\mathcal{A})$ ,  $L(\mathcal{B})$ . Все 4 слова должны быть различными.

#### Задача 4.

Определим язык  $L \subseteq \{a, b\}^*$  индуктивными правилами:

- (1)  $\varepsilon \in L$ ;
- (2) вместе с любым словом  $x \in L$  в  $L$  также входят слова  $xa, xaa, xabba$ ;
- (3) никаких других слов в  $L$  нет.

В язык  $T \subseteq \{a, b\}^*$  входит пустое слово  $\varepsilon$  и ВСЕ начинающиеся и заканчивающиеся буквой  $a$  слова, в которых нет подслов “ $aba$ ” или “ $bbb$ ”. Докажите или опровергните, что  $L = T$ .

1. Докажите или опровергните, что  $L = T$ . Если равенство неверно, то нужно явно указать слово, принадлежащее одному языку и не принадлежащее другому. Если равенство верно, то нужно провести доказательство ПО ИНДУКЦИИ:

- 1)  $L \subseteq T$ ;
- 2)  $T \subseteq L$ .

2. Постройте конечный автомат, распознающий  $T$ . Докажите (по индукции), что построенный автомат распознаёт язык  $T$ .

## 8 Благодарности

Я хотел бы поблагодарить Сергея Тарасова за предоставление своих материалов и задач, часть из которых я использовал при подготовке этих заданий. Также я хочу поблагодарить Дмитрия Гончара за исправление опечаток и рекомендации по правке предыдущей версии этого задания.