# Содержание

# Программа

## 1. Регулярные языки

**1.1.** Введение. Формальные языки, автоматы и грамматики. Их роль в теории вычислений, применение в алгоритмах обработки текстов, приложение к компиляции и другим областям.

**1.2.** Алгебра Клини. Алфавит, слово и язык. Операции конкатенации, объединения, итераций ($*$ и $+$). Подслова, префиксы и суффиксы.

**1.3.** Алгебраическое определение регулярных языков. Регулярные выражения.

**1.4.** Конечные автоматы. Детерминированные конечные автоматы. Построение ДКА по РВ.

**1.5.** Замкнутость регулярных языков относительно теоретико-множественных операций. Конструкция произведения автоматов.

**1.6.** Недетерминированные конечные автоматы. Построение НКА по РВ: позиционный алгоритм и алгоритм пошагового построения по РВ (из книги Хопкрофта-Мотвани-Ульман).

**1.7.** Алгоритм проверки принадлежности слова языку, распознаваемому НКА. Построение ДКА по НКА. Экспоненциальный разрыв между числом состояний НКА и ДКА.

**1.8.** Алгоритмы построения РВ по НКА: алгоритм последовательного удаления состояния, обобщение алгоритма Флойда-Уоршелла.

**1.9.** Конечные автоматы и алгоритмы обработки текста. Алгоритм Кнута-Моррисса-Пратта. Реализация структуры данных «словарь». Алгоритм Ахо-Корасик. Суффиксный автомат. *Суффиксные деревья.

**1.10.** Структурные свойства регулярных языков. Лемма о накачке. Алгоритм минимизации ДКА. Теорема Майхилла-Нероуда.

## 2. Контекстно-свободные языки

**2.1.** Формальные грамматики. Иерархия Хомского. Алгоритмы построения НКА по праволинейной грамматике и ПГ по НКА.

**2.2.** Контекстно-свободные грамматики. Деревья вывода. Грамматики для языков правильных скобочных выражений (языки Дюка) и языка арифметических выражений (формальное определение математических формул). Проблема неоднозначности КС-грамматики. *Существенно-неоднозначные КС-языки.

**2.3.** Автоматы с магазинной памятью. Эквивалентность языков, распознаваемых автоматами с различными условиями приёма. Алгоритм построения МП-автомата по КС-грамматике. Алгоритм построения КС-грамматики по МП-автомату. Детерминированные МП-автоматы.

**2.4.** Свойства замкнутости КС-языков. Замкнутость относительно объединения, пересечения с регулярными языками, морфизма, подстановки, обратного морфизма.

**2.5.** Модифицированный алгоритм Кока-Янгера-Коссами проверки принадлежности слова языку, порождаемому КС-грамматикой (без нормальной формы Хомского). Нормальная форма Хомского: вложенность класса КС-языков в класс контекстно-зависимых языков.

**2.6.** Лемма о накачке. *Теорема Парика.

**2.7***. Определение КС-грамматик через систему уравнений над множествами. Построение РВ по ПГ через решение уравнений с регулярными коэффицентами.

## 3. Элементы синтаксического анализа

**3.1.** Алгоритмы преобразования КС-грамматик: удаления недостижмых нетерминалов и удаления бесплодных нетерминалов.

**3.2.** LL(1)-грамматики и LL(1)-анализаторы. Функции FIRST и FOLLOW. Алгоритм построения LL(1)-анализатора. Критерий LL(1)-грамматики. Удаление левой рекурсии и факторизация.

**3.3.** Рекурсивный спуск. Линейный алгоритм для LL(1)-грамматик

**3.4.** Атрибутные грамматики с синтезируемыми атрибутами. Синтаксически-управляемый перевод на примере преобразования xml в html.

**3.5***. Parsing Expression Grammars

**3.6***. LR(1) и LR(0) -грамматики и -анализаторы. Алгоритмы построения, преобразования LR(1)-анализатора к LR(0)-анализатору в случае LR(0)-грамматики. Критерий LR($k$)-грамматики для $k \leqslant 1$.

## 4*. Неразрешимые задачи в области КС-языков.

**4.1.** Неразрешимость проверки однозначности КС-грамматики. Неразрешимость проверки универсальности КС-грамматики (порождает ли граммаика все слова).

**4.2.** КЗ-грамматики и линейно-ограниченные автоматы. Неразрешимость проблемы пустоты.

**4.3.** Машины Минского. Неразрешимость проблемы пустоты для конечного автомата с двумя счётчиками.

## 5*. Алгоритм Лемпеля–Зива–Велча (LZW) сжатия слов в формализме КС-грамматик и автоматов.

# Литература


[1] *Ахо А., Сети Р., Ульман Дж.* Компиляторы. Принципы, технологии, инструменты. М., СПб., Киев: Вильямс, 2001.

[2] *Мартыненко Б.К.* Языки и трансляции. СПб.: СПбГУ, 2004. Доступно по ссылке http://trpl7.ru/t-books/Martin/Martinenko_FLT_Cont.htm

[3] Теория и реализация языков программирования / В.А. Серебряков, М.П. Галочкин, Д.Р. Гончар, М.Г. Фуругян. Москва: МЗ-Пресс, 2006.

[4] *Хопкрофт Д., Мотвани Р., Ульман Д.* Введение в теорию автоматов, языков и вычислений. Москва: Вильямс, 2002.

[5] *Ахо А., Лам М., Сети Р., Ульман Дж.* Компиляторы. Принципы, технологии и инструментарий. М., СПб., Киев: Вильямс, 2011. 1184 с.

[6] *Рубцов А.А.* Заметки и задачи о регулярных языках и конечных автоматах. Москва: МФТИ, 2019.

[7] *Шень А.* Программирование: теоремы и задачи. 2-е изд. Москва: МЦНМО, 2004. https://www.mccme.ru/free-books/shen/shen-progbook.pdf

[8] *Sipser M.* Introduction to the Theory of Computation. 3rd edn. Cengage Learning, 2012

# Правила оценивания

Итоговая оценка за курс является оценкой за экзамен. Экзамен проходит в письменной форме с возможной последующей устной беседой по решению в процессе показа работ. Также есть следующие варианты получения автомата.

**1. Автомат по двум семестровым контрольным.** Будут проведены две семестровые контрольные. Оценка за каждую — дробное число от 0 до 10 с точностью до сотых. В качестве оценки автоматом можно получить среднюю оценку $O_{\mathbf{KP}}$ за две контрольные после арифметического округления, если выполнены следующие условия. Во-первых, оценка за каждую контрольную должна быть не меньше $3,0$. Во-вторых должно быть сдано домашнее задание. Если задание не сдано полностью — в качестве оценки автоматом можно засчитать оценку $O_{\mathbf{KP}} - 2$, если же задание не сдано частично, то оценку — $O_{\mathbf{KP}} - 1$.

**2. Автомат семинариста.** Студент имеет право засчитать автоматом оценку, предложенную семинаристом (за работу в семестре). Эту оценку также можно засчитать при условии, что каждая из семестровых контрольных написана на оценку не ниже 3.0. В случае, если эта оценка существенно выше $O_{\mathbf{KP}}$, то для её получения возможно потребуется пройти лекторский контроль (беседа с лектором) в результате которого оценка может быть понижена или оставлена без изменений.

В случае, если студент не сдал задание, то он получает две или одну штрафную задачу на экзамене. В случае неверного или неполного решения этих задач от суммы баллов за экзамен будут отняты баллы.

Эти правила могут быть изменены, в случае если случится вторая волна карантина.

# Задание

Задачи, выделенные в дополнительный раздел, а также задачи, помеченные звёздочкой, являются дополнительными и необязательными. Контрольные вопросы являются полноценными задачами, хотя и выделены в отдельные блоки. Решение всех задач должно быть обосновано. Отдельные указания по необходимости обоснования в некторых задачах являются акцентированием и вовсе не означают, что в других задачах обоснование не требуется. Использование алгоритмов из курса (см. программу), считается обоснованием. При использовании алгоритма проверяющий должен иметь возможность проверить корректность протокола: решения в духе «автомат построен по алгоритму, но вот только ответ» не оцениваются.

Если в формулировке вопроса задачи используются обороты «верно ли, что» и «может ли быть», то в случае положительного ответа приведите доказательство, а в случае отрицательного – контрпример. Верное рассуждение без контрпримера оценивается в половину задачи.

Всё вышесказанное относится ко всем письменным работам курса.

# Регулярные языки

**Задача 1.** Вычислить $\{a, a^3, a^5 \ldots\} \cdot \{a, a^3, a^5 \ldots\}$.

**Задача 2.** Построить регулярное выражение (РВ) для

**а)** языка, который содержит все слова, в которых есть как буква $a$, так и буква $b$;

**б)** языка из слов, содержащих в качестве подслова ровно одно слово $ab$;

**в)** языка, слова которого не содержат подслово $ab$;

**Замечание.** В этой задаче необходимо доказать, что построенное РВ порождает требуемый язык. Доказательство корректности является важной частью решения, это относится и ко всем последующим задачам!

**Задача 3.** Постройте ДКА, распознающий язык $\Sigma^* aab \Sigma^*$.

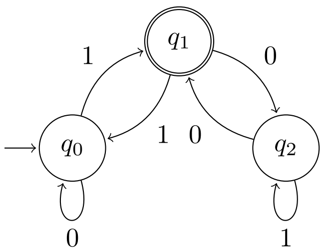
**Задача 4.** Верно ли, что **а)** $\varepsilon \in \{a, aab, aba\}$; **б)** $\varnothing \in \{a, aab, aba\}$?

**Задача 5.** 1. Задайте множество $\{a^n \mid n > 0\} \times \{b^n \mid n \geqslant 0\}$ формулой, которая не использует символ $\times$.

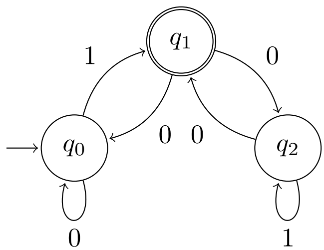
2. Опишите язык $\{a^{3n} \mid n > 0\} \cap \{a^{5n+1} \mid n \geqslant 0\}^*$ регулярным выражением.

**Задача 6.** Автоматы $\mathcal{A}$ и $\mathcal{B}$ заданы диаграммами. Выполните следующие задания.

Автомат $\mathcal{A}$ :




Автомат $\mathcal{B}$ :




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. Явлется ли автомат детерминированным?

Ответьте на вопросы.

3. Опишите последовательность конфигураций автомата $\mathcal{A}$ при обработке слова $w = 011001$. Верно ли, что $w \in L(\mathcal{A})$?

4. Принимает ли автомат $\mathcal{B}$ слово $v = 0101001$?

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

1. $\varepsilon, b, bb \in L$;
2. вместе с любым словом $x \in L$ в $L$ также входят слова $ax, bax, bbax$;
3. никаких других слов в $L$ нет.

Язык $T \subseteq \{a, b\}^*$ состоит из всех слов, в которых нет трёх букв $b$ подряд.

1. Докажите или опровергните, что $L = T$.[1]

2. Запишите язык $T$ в виде регулярного выражения.

3. Постройте конечный автомат, принимающий $T$. Докажите (по индукции), что построенный автомат принимает язык $T$.

**Задача 8.** Выполните следующие задания.

1. Построить ДКА, принимающий язык $L$, состоящий из всех слов в алфавите $\{0, 1\}$, которые содержат чётное число нулей и нечётное число единиц.

2. Построить эквивалентную праволинейную грамматику. Будет ли она однозначной?

3. Построить регулярное выражение для языка $L^R$.

**Задача 9.** Будут ли регулярными следующие языки?

1. $L = \{a^{2020n+5} \mid n = 0, 1, \ldots\} \cap \{a^{503k+29} \mid k = 401, 402, \ldots\} \subseteq \{a^*\}$.

2. $L_2 = \{a^{200n^2+1} \mid n = 1000, 1001, \ldots\} \subseteq \{a^*\}$.

3. Язык $L_3$ всех слов в алфавите $\{0, 1\}$, которые представляют числа в двоичной записи, дающие остаток два при делении на три (слово читается со старших разрядов). Например, $001010 \notin L_3$ ($1010_2 = 10_{10} = 3 \times 3 + 1$), а $10001 \in L_3$ ($10001_2 = 17_{10} = 5 \times 3 + 2$).

**Задача 10.** Порождает ли регулярное выражение $(ab)^*(ba)^*$ тот же язык, что распознаёт ДКА $M = (\{A, B, C, D\},\ \{a, b\},\ \delta,\ A,\ \{A, D, E\})$, где функция переходов задана следующим образом:

$$\delta(A, a) = B,\ \delta(A, b) = C,\ \delta(B, b) = D,\ \delta(C, a) = E,$$

[1] Если равенство неверно, то нужно явно указать слово, принадлежащее одному языку и не принадлежащее другому. Если равенство верно, то нужно провести доказательство по индукции в обе стороны: $L \subseteq T$ и $T \subseteq L$.

$$\delta(D,a)=B,\ \delta(D,b)=C,\ \delta(E,b)=C.$$

**Задача 11.** Покажите, что следующий язык удовлетворяет лемме о разрастании для регулярных языков, но сам регулярным не является:

$$L=\{ab^{2^i}\mid i\geqslant 0\}\cup\{b^j\mid j\geqslant 0\}\cup\{a^m b^n\mid m>1, n\geqslant 0\}.$$

**Задача 12*.** Решите уравнения с регулярными коэффициентами. В каждом пункте нужно выполнить три задания: **а)** найти частное решение; **б)** найти решение, минимальное по включению; **в)** найти все решения.

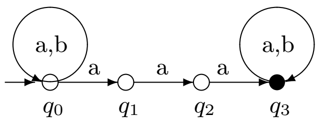
1. $X=((110)^*+111^*)X.$

2. $X=(00+01+10+11)X+(0+1+\varepsilon).$

3. $\begin{cases} Q_0 &= & 0Q_0+1Q_1+\varepsilon,\\ Q_1 &= & 1Q_0+0Q_2,\\ Q_2 &= & 0Q_1+1Q_2.\end{cases}$

**Задача 13.** Автомат $\mathcal{A}_1$ задан диаграммой. Выполните следующие задания. Достаточно выполнить хотя бы один из пунктов 2 или 3.

$\mathcal{A}_1$ :




1. По диаграмме $\mathcal{A}_1$ постройте праволинейную грамматику $G$.

2*. Запишите определяющую систему уравнений для $G$. Найдите её наименьшую неподвижную точку (минимальное по включению решение) и вычислите регулярное выражение $\alpha_1$ для $L(\mathcal{A}_1)$.

3. Определите регулярное выражение $\alpha_2$ для $L(\mathcal{A}_1)$ с помощью индуктивного вычисления множеств $R_{ij}^k$.

4. Выберите регулярное выражение $\alpha_1$ или $\alpha_2$ и постройте по нему НКА $\mathcal{A}_2$.

5. Выберите НКА $\mathcal{A}_1$ или $\mathcal{A}_2$ и постройте по нему ДКА $D_1$.

6. Выберите какое-нибудь регулярное выражение $\alpha_1$ или $\alpha_2$ и постройте ДКА $D_2$.

7. Выберите какой-нибудь ДКА $D_1$ или $D_2$, дополните его, если нужно, до полного и постройте минимальный полный ДКА $\min \mathcal{A}$ для $L$. Для каждой пары состояний укажите соответствующие различающие их цепочки.

8. Постройте КМП-автомат, ищущий вхождение образца $aaa$ в текст и сравните его с $\min \mathcal{A}$.

**Задача 14.** Опишите классы эквивалентности Майхилла–Нероуда для языка $L$ над алфавитом $\Sigma = \{a, b\}$. В случае конечности множества классов постройте минимальный полный ДКА, распознающий $L$, где $L$ — язык **а)** $\Sigma^* ab \Sigma^*$; **б)** $\mathsf{PAL} = \{w \mid w = w^R\}$[2]; **в)** $\{w \mid |w|_{ab} = |w|_{ba}\}$.

**Задача 15.** Язык $L$ состоит из двоичных записей (без ведущих нулей) положительных чисел $n$, входящих в пару $(n, m)$ некоторого решения уравнения $5n + 3m = 17$ в целых числах. Опишите классы эквивалентности Майхилла-Нероуда языка $L$. Является ли язык $L$ регулярным?

Определения КМП-автомата, автомата-словаря и автомата Ахо–Корасик приведены в [6].

**Задача 16.** 1. Постройте КМП-автомат для слова $babbabab$ (над алфавитом $\{a, b\}$).

2. Постройте для того же слова КМП-автомат $\mathcal{A}^{\mathsf{exc}}$ с суффиксными ссылками.

3. Продемонстрируйте работу автомата $\mathcal{A}^{\mathsf{exc}}$ на словах:

**а)** $babbabbabab$; **б)** $babbabc$.

*Под демонстрацией понимается последовательность конфигураций автомата $\mathcal{A}^{\mathsf{exc}}$, т. е. пар из состояния и необработанной части слова.*

**Задача 17.** Постройте ДКА для словаря $\{ac, acb, b, ba, c, cbb\}$. Добавьте в полученный словарь слово $ab$ и удалите слово $ac$.

[2] Здесь $R$ — операция обращения; язык $\mathsf{PAL}$ — это язык палиндромов, т. е. слов, которые читаются справа налево и слева направо одинаково, например «казак».

**Задача 18.** Постройте для словаря $S = \{ac, acb, b, ba, c, cbb\}$ (который вы строили в предыдущей задаче) автомат Ахо–Корасик. Посчитайте с его помощью количество различных вхождений слов из словаря $S$ в слово $acbacbb$ в качестве подслов.

**Задача 19.** Постройте суффиксный автомат $\mathcal{A}$ для слова $abcbc$ и выполните следующие упражнения.

1. Известно, что в тексте (слове) $t$ слово $bcbc$ встретилось 20 раз (как подслово), а слово $bc$ встретилось 60 раз. Сколько могло встретиться слово $cbc$?

2. Постройте минимальный ДКА, распознающий язык $\Sigma^*\mathrm{Suff}(abcbc)$, где $\Sigma = \{a, b, c\}$, а $\mathrm{Suff}(w)$ — множество суффиксов слова $w$.

3. Постройте суффиксный автомат $\mathcal{B}$ для слова $abcbcbc$. Выразите классы эквивалентности Майхилла-Нероуда языка $L(\mathcal{B})$ через классы эквивалентности Майхилла-Нероуда языка $L(\mathcal{A})$ (и операции с языками).

# Контрольные вопросы

Несмотря на название раздела, все решения задач должны быть строго обоснованы.

**Задача 20.** Верно ли, что если пересечение языков $L_1,\ L_2 \subseteq \{a, b\}^*$ содержит язык $F = \{a^n b^n \mid n \geqslant 1\} : F \subseteq L_1 \cap L_2$, то хотя бы один из языков $L_1$ и $L_2$ является нерегулярным?

**Задача 21.** Пусть $X_1, X_2, \ldots, X_n, \ldots$ бесконечное семейство регулярных языков.

1. Верно ли, что язык $X = \bigcup\limits_{n=1}^{\infty} X_i$ является регулярным языком?

2. Верно ли, что язык $X = \bigcap\limits_{n=1}^{\infty} X_i$ является регулярным языком?

**Задача 22.** Язык $L_1$ объединили с конечным языком $R$ и получили язык $L$ ($L = L_1 \cup R$). Язык $L$ оказался регулярным. Верно ли, что язык $L_1$ мог быть нерегулярным?

**Задача 23.** Язык задан контекстно-зависимой грамматикой, которая не является контекстно-свободной. Может ли он быть регулярным?

# Контекстно-свободные языки

**Задача 24.** Язык $L^{=}$ является языком всех слов с равным числом символов $a$ и $b$.

1. Покажите индукцией[1] по длине слова, что КС-грамматика с правилами $S \to SS \mid aSb \mid bSa \mid \varepsilon$ порождает язык $L^{=}$.

$2^{*}$. Грамматика называется линейной, если в правые части правил вывода входит не более одного нетерминала. Покажите, что язык $L^{=}$ не порождается никакой линейной КСГ.

**Задача 25.** Палиндромами называют слова, которые одинаково читаются слева направо и справа налево, например, «ротор».

1. Покажите, что язык палиндромов в произвольном алфавите является КС-языком.

2. Покажите, что дополнение к языку палиндромов (язык всех непалиндромов) также является КС-языком.

**Задача 26.** Покажите, что дополнение языка $U = \{a^n b^n c^n \mid n \geqslant 0\}$ является КС-языком.[2]

**Задача 27.** Являются ли следующие языки КС-языками:

**а)** $SQ = \{ww \mid w \in \{a,b\}^*\}$; **б)**$^\circ$ $\Sigma^* \setminus SQ$ **в)** $\{a^{3^n} \mid n > 0\}$?

**Задача 28.** Выполните следующие задания.

1. Постройте магазинный автомат (МА), распознающий язык $L^{=}$ из задачи 24.

2. Постройте детерминированный МА, распознающий язык $L^{=}$ (достаточно выполнить только этот пункт).

**Задача 29.** Язык Дюка с двумя типами скобок $D_2$ порождается грамматикой

$$S \to SS \mid (S) \mid [S] \mid \varepsilon.$$

[1] Другие доказательства, кроме индукции, не принимаются.

[2] Так как сам язык $U$ не является КС-языком, то это означает, что в отличие от регулярных языков множество КС-языков не замкнуто относительно дополнения.

1. Постройте недетерминированный МП-автомат, распознающий язык $D_2$.

2. Постройте детерминированный МП-автомат, распознающий язык $D_2$ (достаточно выполнить только этот пункт).

**Задача 30.** Для языка

$$L = \{w \mid w = xcy; x, y \in \{a, b\}^*; |x| = |y|\}$$

**а)** постройте КС-грамматику $G$, порождающую язык $L$;

**б)** постройте недетерминированный МА, эквивалентный этой грамматике;

**в)** продемонстрируйте работу построенного МА на слове $acab$ (проанализируйте все варианты поведения).

**Задача 31.** Заданы грамматика $G = \{\ \{\ A,\ B,\ C,\ D,\ E,\ F,\ S\ \},\ \{a, b\},$ $\{S \to AB \mid C,\ A \to aE \mid a,\ E \to aE \mid \varepsilon,\ B \to bB \mid Bb \mid b,\ C \to CD,$ $F \to ab,\ D \to aba\},\ \ S\}$ и магазинный автомат $M = (\{q_0\},\ \{a, b\},$ $\{\ S,\ a,\ b,\ A,\ B\ \},\ \{\ \delta(q_0, \varepsilon,\ S) = \{(q_0, AB)\},\ \delta(q_0, \varepsilon,\ A) = \{(q_0, aA), (q_0, a)\},$ $\delta(q_0, \varepsilon,\ B) = \{(q_0, bB), (q_0, b)\}, \delta(q_0, a, a) = \{(q_0, \varepsilon)\},\ \delta(q_0, b, b) = \{(q_0, \varepsilon)\},$ $q_0,\ S\}$, принимающий слова опустошением магазина.

1. Эквивалентны ли грамматика $G$ и $N$-автомат[3] $M$?

2. Однозначна ли грамматика $G$? Если нет, то постройте эквивалентную ей однозначную грамматику.

3. Является ли автомат $M$ детерминированным? Если нет, постройте эквивалентный ему детерминированный МА.

**Задача 32.** Определим языки $L_1 = \Sigma^* aab \Sigma^*$, где $\Sigma = \{a, b\}$, и

$$L = \{w \mid w \in \overline{L_1},\ |w|_a \geqslant |w|_b\}.$$

1. Является ли дополнение языка $L$ КС-языком?

2. Является ли дополнение языка $L$ регулярным языком?

[3] Мы называем $N$-автоматом МП-автомат, допускающий по пустому стеку, а $F$-автоматом — МП-автомат, допускающий по принимающему состоянию.

**Задача 33.** Язык $L$ задан КС-грамматикой с правилами:

$$S \to aSa \mid aSb \mid bSa \mid bSb \mid a.$$

1. Является ли $L$ регулярным языком?
2. Является ли дополнение $L$ регулярным языком?
3. Является ли $L$ КС-языком?
4. Является ли дополнение $L$ КС-языком?

**Задача 34.** Язык $L$ задан КС-грамматикой с правилами:

$$S \to aSb \,|\, A \,|\, B \,|\, \varepsilon, \quad A \to aAa \,|\, \varepsilon, \quad B \to bBb \,|\, \varepsilon.$$

1. Является ли $L$ регулярным языком?
2. Является ли дополнение $L$ регулярным языком?
3. Является ли $L$ КС-языком?
4. Является ли дополнение $L$ КС-языком?

# Контрольные вопросы

**Задача 35.** КС-грамматика называется *левооднозначной*, если каждое слово порождаемого ею языка имеет единственный левый вывод. Аналогично определяется *правооднозначная грамматика.* Можно ли построить пример левооднозначной, но не правооднозначной КС-грамматики?

**Задача 36.** Известно, что $L_1$ — КС язык, не являющийся регулярным, а $L_2$ — не КС-язык. Может ли язык $L_2L_1$ быть регулярным языком? При положительном ответе привести пример.

# Приложение КС-грамматик для сжатия данных*

Некоторые алгоритмы сжатия строк можно описать в терминах КС-грамматик. Мы рассмотрим два таких алгоримта. Первый из них носит названеие «Straight-line program» (SLP) и состоит в следующем. Слово $w$ описывают с помощью КС-грамматики $G_w$, которая порождает единственное слово: $L(G_w) = w$. Грамматику $G_w$ называют «Straight-line grammar» (SLG); этим же термином иногда называют и описываемый нами частный случай метода сжатия SLP: в роли программ выступают КС-грамматики.

**Пример 1.** Грамматика, описываемая правилами

$$S \to A_1 A_1,\ \ A_1 \to A_2 A_2,\ \ A_2 \to A_3 A_3,\ \ \ldots,\ \ A_{n-1} \to A_n A_n,\ \ A_n \to a$$

порождает единственное слово $a^{2^n}$. Длина описания грамматики не превосходит $cn$, для некоторой константы $c > 0$, то есть имеет длину порядка логарифма от длины порождаемого слова, что является хорошим коэффициентом сжатия.

**Задача 37.** Постройте SLG $G_n$, порождающую слово

$$a^n b a^{n-1} b a^{n-2} b \ldots ababa^2 ba^3 b \ldots a^n b.$$

Длина описания $G_n$ должна быть $cn, c > 0$. В качестве решения можно построить SLG $G_5$.

**Замечание 1.** *Преимуществом описанного метода сжатия является возможность эффективной проверки сжатого слова на регулярные события без разархивации. То есть, существует алгоритм, получающий на вход описание НКА $\mathcal{A}$ и SLP $G_w$ и проверяющий непустоту пересечения $L(\mathcal{A}) \cap L(G_w)$ за полиномиальное время от длин описаний $\mathcal{A}$ и $G_w$, но не $w$.*

**Задача 38*.** Постройте описанный выше алгоритм и докажите его корректность.

Мы описали общий метод сжатия SLP, но не описали пока алгоритма сжатия строк в грамматики. Таких алгоритмов существует несколько, одним из популярных алгоритмов сжатия такого типа является алгоритм Лемпеля-Зива-Велча (Lempel-Ziv-Welch, LZW). Опишем работу этого алгоритма на примере сжатия конкретной строки: $aababbbbaabaabab$.

| $a$ | $ab$ | $abb$ | $b$ | $ba$ | $aba$ | $abab$ |
|---|---|---|---|---|---|---|
| $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ |

Таблица 1. разбиение строки алгоритмом LZW

Таблица 1 представляет собой словарь. Она устанавливает взаимно однозначное соответствие между нетерминалами и словами: слово $w_i$ в построенной в итоге грамматике будет выводимо из $A_i$ и только из $A_i$ (но не обязательно за один шаг вывода). Опишем алгоритм заполнение таблицы-словаря.

1. В начале работы словарь пуст, слово $u$ – необработанный суффикс слова $w$ – совпадает с $w$, $i = 1$.

2. Алгоритм ищет максимальный префикс $x$ необратонной части входа $u$, который был добавлен в словарь.

3. Если $u = xav, a \in \Sigma$, то алгоритм добавляет в словарь слово $w_i = xa$, удаляет префикс $xa$ из $u$, увеличивает $i$ на 1 и переходит к предыдущему шагу, если $u \neq \varepsilon$. Если же $u = \varepsilon$, алгоритм заканчивает работу.

4. Если $u = x$ и $x$ уже соответствует некоторому нетерминалу $A_j$, то алгоритм добавляет в грамматику правило $A_i \to A_j$ и завершает работу. Обратим внимание, что $x$ на этом шаге является суффиксом $w$.

Так, первая буква слова $w$ всегда будет приписана нетерминалу $A_1$; далее в нашем примере за первой $a$ идёт подслово $ab$, которое приписывается нетерминалу $A_2$, поскольку первая буква подслова $a$ уже была приписана $A_1$; далее идёт подслово $abb$ – подслово $ab$ уже было преписано $A_2$ и т.д.

Нетрудно заметить, что искомая SLG имеет вид

$$S \to A_1 A_2 \dots A_7, \quad A_1 \to a, \quad A_2 \to A_1 b, \quad A_3 \to A_2 b,$$

$$A_4 \to b, \quad A_5 \to A_4 a, \quad A_6 \to A_2 a, \quad A_7 \to A_6 b.$$

Но как её эффективно построить алгоритмически, равно как и таблицу 1? Для этого воспользуемся техникой, базирующейся на конечных автоматах.

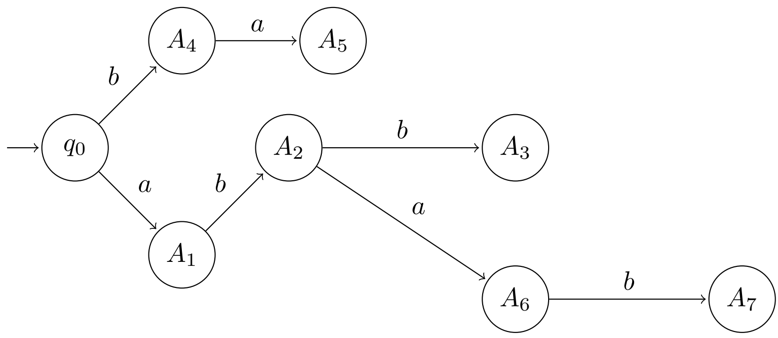



Рис. 2. LZW-автомат

В процессе построения SLG по алгоритму LZW мы строим LZW-автомат (рис. 2), который по-сути реализует словарь. Однако помимо стандартных функций словаря, LZW-автомат помечает каждую вершину, кроме начальной, нетерминалом $A_i$, устанавливая тем самым соответсвие между словом $w_i$ и состоянием автомата: $q_0 \xrightarrow{w_i} A_i$.

Итак, опишем алгоритм LZW построения SLG. В начале работы алгоритма словарь (реализуемый LZW-автоматом) пуст; обозначим через $u$ необработанную часть входа – в начале работы $u = w$.

Алгоритм находит кратчайший префикс $x \neq \varepsilon$ слова $u = xy$, которого ещё нет в словаре и добавляет его в словарь, помечая вершину, соответсвующую этому слову новым нетерминалом $A_i$. Алгоритм повторяет этот процесс удалив из $u$ префикс $x$ $(u = y)$ до тех пор пока либо слово $u$ не окажется пустым, либо $u$ не будет содержаться в словаре. При этом, если префикс $x$ добавляется в словарь, то $x = va, a \in \Sigma$, а слово $v$ уже было добавлено в словарь и ему соответствует некоторый нетерминал $A_j$. Тогда алгоритм добавляет в грамматику переход $A_i \to A_j a$. Если же $u$ целиком содержится в словаре, то ему уже соответствует нетерминал $A_j$ – в этом случае алгоритм добавляет правило $A_i \to A_j$. После окончания построения LZW-автомата, алгоритм добавляет к грамматике правило $S \to A_1 \dots A_n$, где $n$ – номер последнего добавленного нетерминала.

Приведённый алгоритм очевидно работает за линейное время (от длины $w$). Строку, сжатую алгооритмом LZW легко декодировать как и строку, заданную произвольной SLG: нужно вывести единственную строку из грамматики, при этом каждый нетерминал раскрывается единственным образом. Также для алгоритма LZW справедливо замечание 1.

**Задача 39.** Постройте LZW-автомат и SLG $G_w$ по описанному выше алгоритму для слова $w$:

**а)** $w = a^8$;  1. $w = tobeornottobeortobeornot$.

**Задача 40.** Постройте для слова $w = tobeornottobeortobeornot$ SLG, которая оптимальнее, чем построенная по алгоритму LZW. Численным показателем оптимальности является сумма длин правых частей всех правил SLG.

# Элементы синтаксического анализа

## LL-анализ

**Задача 41.**

1. Определите, какие из грамматик ниже являются LL(1)-грамматиками.

2*. Определить, являются ли LL($k$)-грамматиками[1] следующие грамматики (заданные правилами). Если да, указать точное значение $k$:

а) $S \to Ab$, $A \to Aa \mid a$;

б) $S \to Ab$, $A \to aA \mid a$;

в) $S \to aAb$, $A \to BB$, $B \to ab \mid A \mid \varepsilon$;

г) $S \to aAb$, $A \to AaAb \mid \varepsilon$;

д) $S \to aB$, $B \to aBB \mid b$.

**Задача 42.** Построить LL(1)-грамматику, эквивалентную грамматике из задачи 41(б), и управляющую таблицу для неё.

**Задача 43.** Написать для грамматики эквивалентную LL(1)-грамматику, построить LL(1)-анализатор и продемонстрировать его работу на слове $baab$.

$$S \to baaA \mid babA \qquad A \to \varepsilon \mid Aa \mid Ab$$

[1] Формальное определение приведено в [3]

**Задача 44*.** Докажите, что язык $a^* \cup a^n b^n$ не является LL(1)-языком, то есть не существует LL(1)-грамматики, порождающей этот язык.

**Задача 45.** Язык $L$ задан неоднозначной КС-грамматикой

$$G = \{\{S\}, \{(,)\}, \{S \to (S) \mid SS \mid ()\}, S\}.$$

Написать LL(1)-грамматику для языка $L$.

## Контрольные вопросы

**Задача 46.** Существует ли такая праволинейная (не обязательно регулярная праволинейная) грамматика, которая не является LL(1)-грамматикой?

**Задача 47.** В приведённой грамматике[2] $G$ есть правило $S \to AB$ и при этом $\mathrm{FIRST}(A) \cap \mathrm{FIRST}(B) = \varepsilon$. Верно ли, что грамматика $G$ может быть LL(1)-грамматикой?

**Задача 48.** Пусть для некоторых двух нетерминалов $A$ и $B$ приведённой КС-грамматики $G$ пересечение $\mathrm{FOLLOW}(A) \cap \mathrm{FOLLOW}(B)$ оказалось непустым. Верно ли, что грамматика $G$ не является LL(1)-грамматикой?

# LR-анализ*

**Задача 49.** Дана грамматика $G = \{\ \{A, S\},\ \{a, b, c\},\ \{\ S \to Aa \mid b \mid \varepsilon;\ A \to Ab \mid c\ \},\ S\ \}$. Является ли грамматика $G$ LR($k$)-грамматикой? При положительном ответе на вопрос найти минимальное $k$ и построить соответствующий анализатор. Построить дерево разбора для цепочки $cbba$.

**Задача 50.** Дана грамматика $G = \{\ \{A, S\},\ \{a, b, c\},\ \{S \to Aa \mid b;\ A \to\ Ab \mid c\ \},\ S\ \}$. Является ли грамматика $G$ LR($k$)-грамматикой?

[2] Грамматика называется приведённой, если в ней нет недостижимых и бесплодных символов. В литературе также встречаются неэквивалентные определения этого термина.

| I | a | $ | + | * | S' | S | T | Q | F | W | + | a | * | $ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $I_0$ | S | | | | | 1 | 2 | | 3 | | | 4 | | |
| $I_1$ | | R (S'->S) | | | | | | | | | | | | |
| $I_2$ | | R (Q->) | S | | | | | 5 | | | 6 | | | |
| $I_3$ | | R (W->) | R (W->) | S | | | | | | 7 | | | 8 | |
| $I_4$ | | R (F->a) | R (F->a) | R (F->a) | | | | | | | | | | |
| $I_5$ | | R (S->TQ) | | | | | | | | | | | | |
| $I_6$ | S | | | | | | 9 | | 3 | | | 4 | | |
| $I_7$ | | R (T->FW) | R (T->FW) | | | | | | | | | | | |
| $I_8$ | S | | | | | | | | 10 | | | 4 | | |
| $I_9$ | | R (Q->) | S | | | | | 11 | | | 6 | | | |
| $I_{10}$ | | R (W->) | R (W->) | S | | | | | | 12 | | | 8 | |
| $I_{11}$ | | R (Q->+TQ) | | | | | | | | | | | | |
| $I_{12}$ | | R (W->*FW) | R (W->*FW) | | | | | | | | | | | |

Рис. 3. автомат LR(1)-анализатора

При положительном ответе на вопрос найти минимальное $k$ и построить соответствующий анализатор. Продемонстрировать работу анализатора на цепочке $cbbab$.

**Задача 51*.** Дана грамматика $G = \{\ \{A, S\},\ \{a\},\ \{\ S \to A;\ A \to aAa \mid a\ \},\ S\}$. Является ли грамматика $G$ LR($k$)-грамматикой? При положительном ответе на вопрос найти минимальное $k$ и построить соответствующий анализатор. Построить дерево разбора для цепочки $aaaaa$.

**Задача 52.** На рис. 3 приведен автомат LR(1)-анализатора (запись $A \to$ обозначает правило $A \to \varepsilon$). В приведенной ниже конфигурации LR-анализатора в первой компоненте (содержимом магазина) опущены состояния автомата. В процессе разбора строки $z \in L(G)$ автомат оказался в конфигурации $\langle cF * FW, +a + a \rangle$. Требуется:

1. Восстановить состояния автомата в содержимом магазина.

2. Восстановить какую-либо из возможных строк $z \in L(G)$, разбор которой мог привести к этой конфигурации.

3. Продемонстрировать процесс разбора на этой строке. Решение обоснуйте.

**Задача 53.** Зафиксируем КС-грамматику $G$ и рассмотрим множество её LR(0)-ситуаций. Будем говорить, что между двумя ситуациями $[\alpha.X\beta]$ и $[\alpha X.\beta]$ определён переход по $X \in N \cup T$; также между ситуациями $[A \to \alpha.B\beta]$ и $[B \to .\gamma]$ определён $\varepsilon$-переход.

Конечный автомат, состояниями которого являются LR(0)-ситуации, а переходы определены по правилам, указанным выше, называют (недетерминированным) LR(0)-*автоматом* или (недетерминированным) *автоматом Кнута.* Автомат полученный в результате детерминизации описанного автомата называют детерминированным LR(0)-автоматом или детерминированным автоматом Кнута.

1. Выпишите все LR(0)-ситуации для грамматики $G$, заданной правилами $S \to aS \mid b$.

2. Постройте недетерминированный автомат Кнута для грамматики $G$.

3. Постройте детерминированный автомат Кнута для грамматики $G$.

4. Постройте LR(0)-анализатор для грамматики $G$. Сравните автомат Кнута с таблицей переходов LR(0)-анализатора для грамматики $G$.

**Задача 54.** Грамматика $G$ задана правилами:

$$S \to Ab, \quad A \to aAa, \quad A \to B, \quad B \to b.$$

1. Построить LR(1) и LR(0)-анализаторы для грамматики $G$ по алгоритму из курса.

2. Постройте LR(0)-анализатор по LR(1)-анализатору из пункта 1 следующим образом. Сотрите все аванцепочки и постройте управляющую таблицу LR(0)-анализатора по получившемуся автомату Кнута. Верно ли, что полученный LR(0)-анализатор является аналиазтором для грамматики $G$? То есть для любого слова, порождаемого $G$, анализатор строит корректный правый разбор, а слова, не порождаемые $G$, анализатор отвергает.

## Контрольные вопросы

**Задача 55.** При построении LR(1)-анализатора для грамматики $G$ в одном множестве оказались ситуации $[A \to .aA\alpha, b]$ и $[B \to \beta.a, a]$, где $\alpha, \beta$ некоторые цепочки из $(N \cup T)^*$. Может ли грамматика $G$ оказаться LR(0)-грамматикой?

# Атрибутные грамматики

Следующий за синтаксическим анализом этап в процессе компиляции является генерация кода. В основе этого этапа лежат вычисления по дереву разбора, которые описывают с помощью атрибутных грамматик. Мы не будем детально изучать эту тему, а изучим лишь частный случай атрибутных грамматик (с синтезируемыми атрибутами).

**Определение 1.** КС-грамматика $G$ называется *атрибутной с ситнезируемыми атрибутами*, если каждому нетерминалу поставлен в соответствуие набор переменных (атрибутов), и при этом для каждого правила грамматики

$$X_0 \to u_0 X_1 u_1 X_2 \dots u_{n-1} X_n u_n,\ X_i \in N, u_i \in \Sigma^*$$

Задан набор правил вычисления некоторых атрибутов

$$X_0[\mathsf{attr}] = f(v_1, v_2, \dots, v_m),$$

в котором $v_i = X_j[\mathsf{attr}_k]$ – значение атрибута $\mathsf{attr}_k$ нетерминала $X_j$, а $f$ – некоторая функция (которая зависит от некоторых атрибутов некторых нетерминалов из правой части правила). Более формально:

$$\begin{aligned} X_0[\mathsf{attr}] = f(&X_1[\mathsf{attr}_{1,1}], X_1[\mathsf{attr}_{1,2}], \dots, X_1[\mathsf{attr}_{1,m_1}],\\ &X_2[\mathsf{attr}_{2,1}], X_2[\mathsf{attr}_{2,2}], \dots, X_1[\mathsf{attr}_{2,m_2}],\\ &\dots\\ &X_n[\mathsf{attr}_{n,1}], X_n[\mathsf{attr}_{n,2}], \dots, X_n[\mathsf{attr}_{n,m_n}]), \end{aligned}$$

Набор правил вычислений атрибутов называют *атрибутной схемой.*

**Пример 2.** Грамматика $G$ задана правилами

$$S \to 1D \mid 0,\ D \to 1D \mid 0D \mid 1 \mid 0$$

и порождает язык двоичных записей натуральных чисел. Определим атрибутную схему для этой грамматики

$$\begin{array}{llll} S \to 0 & D \to 1 & D \to 0 & S \to 1D \\ S[\mathsf{val}] = 0 & D[\mathsf{val}] = 1 & D[\mathsf{val}] = 0 & S[\mathsf{val}] = D[\mathsf{ord}] + D[\mathsf{val}] \\ & D[\mathsf{ord}] = 2 & D[\mathsf{ord}] = 2 & \end{array}$$

$$\begin{array}{ll} D_0 \to 1D_1 & D_0 \to 0D_1 \\ D_0[\mathsf{val}] = D_1[\mathsf{ord}] + D_1[\mathsf{val}] & D_0[\mathsf{val}] = D_1[\mathsf{val}] \\ D_0[\mathsf{ord}] = 2 \times D_1[\mathsf{ord}] & D_0[\mathsf{ord}] = 2 \times D_1[\mathsf{ord}] \end{array}$$

В случае, если правило содержит несколько одинаковых нетерминалов мы нумеруем их вхождение и различаем атрибуты как в случае двух последних правил. Нетерминал $S$ имеет единственный атрибут val, а нетерминал $D$ – атрибуты val и ord. Приведённая атрибутная схема вычисляет значение числа по его двоичной записи. Атрибут ord равен $2^l$, где $l$ – длина слова выведенного из $D$, атрибут val равен значению числа, двоичная запись которого выведена из нетерминала.

Приведём пример вычисление атрибутов для слова 1101. Атрибуты вычисляются снизу вверх.

**Задача 56.** Грамматика RExpr $= \langle\{E, T\}, \{a, b, +, ^*, (, )\}, P, E\rangle$ имеет множество правил $P$ :

$$E \to T + E \mid T, \qquad T \to CT \mid C, \qquad C \to (E) \mid C^* \mid a \mid b.$$

и порождает регулярные выражения над алфавитом $\{a, b\}$.

1. Постройте для грамматики RExpr LR(1)-анализатор[3].

2. Дополните грамматику RExpr до атрибутной так, чтобы она вычисляла атрибуты firstpos, lastpos и nullable согласно алгоритму их вычисления при построении ДКА по РВ. Считайте атрибуты firstpos, lastpos и nullable у терминалов заданными: на практике перед вычислением атрибутов происходит препроцессинг, во время которого могут быть заданы атрибуты терминалов, если это требуется.

[3] Анализатор для этой грамматики довольно громоздкий. Постройте его с помощью программы, например на сайте http://lrk.umeta.ru, используйте программу для выполнения последующих пунктов.

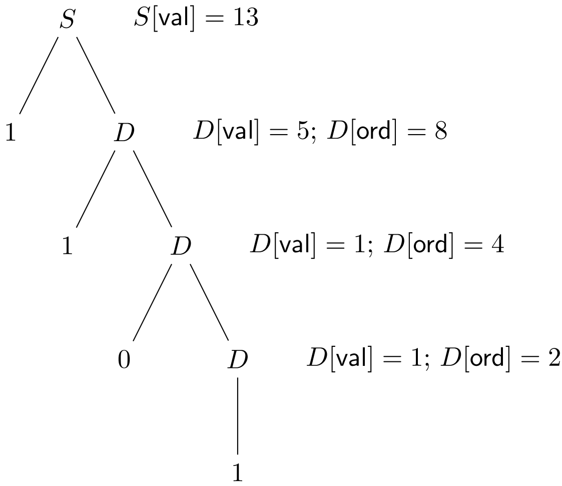



Рис. 4. вычисление атрибутов.

Указание. Вычисление атрибута нужно определять через функцию, которую можно описать на языке программирования или псевдокоде. Пример вычисления атрибута nullable для правила $E_0 \to T + E_1$:

```
E0[nullable] = function( T[nullable], E1[nullable]){
    if( T[nullable] or E1[nullable]){
        return True;
    }else{ return False; }
}
```

3*. Добавьте в атрибутную схему вычисление атрибута followpos.

4. С помощью анализатора постройте дерево разбора для РВ $(a+ab)^*+ab^*$ и вычислите атрибуты firstpos, lastpos, nullable (*followpos) согласно атрибутной схеме (предварительно задав атрибуты у терминалов).

Язык разметки web-страниц HTML был разработан так, что бы его код было легко разбирать интерпретатором. Код на HTML представляет собой правильное скобочное выражение, в котором скобки имеют имена и называются тегами: <tag> ... </tag>. Внутри открывающей скобки <tag> могут быть атрибуты; в общем случае (открывающий) тег имеет вид <name attr1="value" attr2="value" ... attrN=" value">, т. е. сначала идёт имя тега, а потом перечисляются атрбиуты.

При интерпретации HTML-кода правильная скобочная последовательность тегов интерпретируется как дерево: если один тег вложен в другой, то внешний тег — родитель внутреннего. Так, в случае кода

<tag1> ... <t2>....</t2> <t3>....</t3>...</tag1>

тег tag1 является родителем тегов t2 и t3.

Рассмотрим тег <div>, который является контейнером для разделения содержимого страницы. Будем рассматривать два атрибута: align и style. В случае вложенных тегов, атрибут align либо задан, например <div align="center">, либо наследуется от родителя. Атрибут style позволяет задавать (CSS) стиль элемента и устроен более сложно, чем align. Мы сосредоточимся только на задании с помощью атрибута style цвета фона у контейнера <div>. В случае вложенных <div>, если цвет фона не задан, то он наследуется у родителей.

**Задача 57.** Постройте по коду на рис. 5 дерево html документа. Определите значение атрибута align у каждого из узлов div дерева, а также определите цвет фона элемента. Проверьте себя, сохранив текст ниже в файле с расширением .html и открыв файл в браузере.

**Комментарий.** В HTML-документе код документа окружается тегом <html>, внутри тега <head> находятся вспомогательные данные (такие как заголовки), а содержимое документа находится в теге <body>. Внутри тега <style> описывается стиль элементов документа, в нашем коде там указан базовый стиль для тегов <div>: наличие границы, отступы и размер в процентах относительно размера тега-родителя. Браузер интерпретирует документ HTML как дерево, точнее модель документа называется DOM (Document Object Model).

# Дополнительные задачи

В этот раздел входят задачи для подготовки к контрольным работам и экзаменам, а также задачи повышенной сложности для студентов, претендующих на высокие оценки. Задачи данного раздела не являются обязательными для прохождения процедуры сдачи задания, если только не входят в требования семинариста. Во всех письменных общекурсовых работах значение $k$ в задачах на построение LR($k$)-анализаторов не превосходит единицу.

```
<html>
<head>
  <style>
   div{border: 1px solid  black;  padding:1px;
        margin: 1px; width:40%; height:40%;}
  </style>
</head>
<body>
<div style="background-color:lightblue; width:500px;
       height:500px;" align="center">1
  <div style="background-color:blue;" align="left">
    2
    <div align="right">
      3
      <div style="background-color:gray;" align="center">
        4
      </div>
    </div>
    <div>
      5
    </div>
  </div>
  <div>
    6
  </div>
</div>
</body>
</html>
```

Рис. 5. HTML-код для задачи 57

# Регулярные языки

**Задача 58.** Пусть $X$ регулярный язык. Верно ли, что язык $\bigcap\limits_{n=1}^{\infty}(\Sigma^* \setminus X)^n$ является регулярным?

**Задача 59.** Приведите пример бесконечного регулярного языка $X \subset$ $\subset \{a, b\}^*$, отличного от множества всех слов, такого что $X \cap (\Sigma^* \setminus X)^R =$ $X$.

**Задача 60.** Найдите разбиения на минимальное число классов правоинвариантной (И/ИЛИ левоинвариантной) эквивалентности, которые индуцируют следующие языки.

1. Язык, порождаемый выражением $00(10 + 01)^*$.

2. Язык $\{a^{n^2} \mid n \geqslant 0\}$ в однобуквенном алфавите.

# КС-языки

**Задача 61.** Язык $L$ задан грамматикой $G$:

$$S \to bSa \mid AB \mid \varepsilon, \quad A \to bAb \mid b, \quad B \to aBa \mid \varepsilon.$$

Является ли язык $L$ и его дополнение **а)** регулярным языком;

**б)** КС-языком?

**Задача 62.** Являются ли следующие языки КС-языками?

1. $\{x \mid x \in \{c, b\}^*, |x|_c = |x|_b, \forall u, v : x = uv, |u| \neq 0, |v| \neq 0, |u|_c > |u|_b\}$.
2. $\{a^{3^n} \mid n > 0\}$.

**Задача 63*.** Пусть – МА. Построите МА $B$, принимающий все префиксы языка $L(A)$, т.е. язык $L(B) = \{x \mid \exists y : xy \in L(A)\}$ .

**Задача 64.** Для языка

$$L = \{w \mid w = xc^{3k}y; x, y \in \{a, b\}^*; |xy|_a = 2n;\ n, k \geqslant 0\}$$

$$( \ |xy|_a \text{ – число символов } a \text{ в слове } xy)$$

**а)** постройте КС-грамматику $G$, порождающую язык $L$;

**б)** постройте недетерминированный МА, эквивалентный этой грамматике;

**в)** продемонстрируйте работу построенного МА на слове $accab$ (проанализируйте все варианты поведения).

**Задача 65.** Заданы языки $L_1 = \{a^n b^n c^m : n \geqslant 1, m \geqslant 0\}$, $L = \{f^n a^m b^m : n \geqslant 0, m \geqslant 0\}$. Для языка $L_1 \cup L_2$ построить однозначную КС-грамматику и детерминированный МП-автомат. Решение обосновать.

# Элементы синтаксического анализа

**Задача 66.** Язык $L$ задан неоднозначной КС-грамматикой:

$$G = \{\, \{S\},\, \{a, ., ^\wedge, [\,, ]\,, (\,, )\,\},\, \{\, S \to a \mid S.S \mid S[S] \mid S^\wedge \mid S(S)\,\},\, S\,\}.$$

Написать LL(1)-грамматику для языка $L$.

**Задача 67*.** Дана грамматика $G = \{\ \{A, B, C, D, E, S\},\ \{a, b\},\ \{S \to AB,\ A \to a,\ B \to CD \mid aE,\ C \to ab,\ D \to bb,\ E \to bba\ \},\ S\ \}$. Является ли грамматика $G$ LR($k$)-грамматикой? При положительном ответе на вопрос найти минимальное $k$ и построить соответствующий анализатор. Продемонстрировать работу анализатора на цепочке $aabbb$.

Задание содержит авторские задачи педагогического коллектива кафедры МОУ и классические задачи теории формальных языков.

С методическими материалами по курсам кафедры МОУ можно ознакомиться на страницах:

http://www.mou.mipt.ru, http://lrk.umeta.ru, http://rubtsov.su.