МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. ЛОМОНОСОВА

Механико-математический факультет
Филологический факультет

А. Е. Пентус, М. Р. Пентус

Теория формальных языков

Учебное пособие

Москва 2004

УДК 519.713
ББК 28.25
П25

Рецензенты:
Доктор физико-математических наук В. А. Успенский,
Кандидат физико-математических наук В. Н. Крупский

Печатается по постановлению
Редакционно-издательского совета
филологического факультета МГУ

Пентус А. Е., Пентус М. Р.
П25 Теория формальных языков: Учебное пособие. — М.: Изд-во ЦПИ при механико-математическом ф-те МГУ, 2004. — 80 с.

Учебное пособие посвящено классическому разделу математической лингвистикой и теоретической информатики — теории формальных языков. Рассматриваются порождающие грамматики, классификация формальных языков по Хомскому, регулярные выражения, конечные автоматы, автоматы с магазинной памятью, алгоритмические проблемы, связанные с контекстно-свободными грамматиками.

Для студентов, аспирантов и специалистов, занимающихся математической лингвистикой или теоретической информатикой.

УДК 519.713
ББК 28.25

Введение

Учебное пособие содержит основные определения и теоремы курса по теории порождающих грамматик и формальных языков, рассчитанного на 16 теоретических занятий по два академических часа. Материал тщательно структурирован. Факультативные разделы и пункты помечены звёздочками.

В пособии приведены главным образом теоретические результаты. Развёрнутые доказательства, примеры и приложения можно найти в других книгах, ссылки на которые имеются в каждом разделе.

Многие определения и результаты пояснены простыми примерами. Из примера, приведённого сразу после леммы или теоремы, часто можно понять идею доказательства.

Изложение строго математическое, но в то же время используются только самые простые математические понятия. Пособие можно рекомендовать студентам математических, лингвистических и компьютерных специальностей.

1. Слова, языки и грамматики

1.1. Формальные языки

[Гин, с. 12–14], [АхоУль, 0.2], [Сал, 1.1], [Гла, 1.1], [ХопМотУль, 1.5], [ГорМол, с. 347–349], [СокКушБад, с. 11–12], [LewPap2, 1.7], [Рей, с. 22–23], [КукБей, с. 257–262], [АхоСетУль, 3.3]

Определение 1.1. Будем называть *натуральными числами* неотрицательные целые числа. Множество всех натуральных чисел $\{0, 1, 2, \ldots\}$ обозначается $\mathbb{N}$.

Определение 1.2. *Алфавитом* называется конечное непустое множество. Его элементы называются *символами* (*буквами*).

Определение 1.3. *Словом* (*цепочкой*, *строкой*) (string) в алфавите Σ называется конечная последовательность элементов Σ.

Пример 1.4. Рассмотрим алфавит $\Sigma = \{a, b, c\}$. Тогда $baaa$ является словом в алфавите Σ.

Определение 1.5. Слово, не содержащее ни одного символа (то есть последовательность длины 0), называется *пустым словом* и обозначается ε.

Определение 1.6. *Длина* слова w, обозначаемая $|w|$, есть число символов в w, причём каждый символ считается столько раз, сколько раз он встречается в w.

Пример 1.7. Очевидно, $|baaa| = 4$ и $|\varepsilon| = 0$.

Определение 1.8. Если x и y — слова в алфавите Σ, то слово xy (результат приписывания слова y в конец слова x) называется *конкатенацией* (*катенацией*, *сцеплением*) слов x и y. Иногда конкатенацию слов x и y обозначают $x \cdot y$.

Определение 1.9. Если x — слово и $n \in \mathbb{N}$, то через x^n обозначается слово $\underbrace{x \cdot x \cdot \ldots \cdot x}_{n \text{ раз}}$. По определению $x^0 \rightleftharpoons \varepsilon$ (знак $\rightleftharpoons$ читается "равно по определению"). Всюду далее показатели над словами и символами, как правило, являются натуральными числами.

Пример 1.10. По принятым соглашениям, $ba^3 = baaa$ и $(ba)^3 = bababa$.

Определение 1.11. Множество всех слов в алфавите Σ обозначается Σ^*.

Определение 1.12. Множество всех непустых слов в алфавите Σ обозначается Σ^+.

Пример 1.13. Если $\Sigma = \{a\}$, то $\Sigma^+ = \{a, aa, aaa, aaaa, \ldots\}$.

Определение 1.14. Говорят, что слово x — *префикс* (*начало*) слова y (обозначение $x \sqsubset y$), если $y = xu$ для некоторого слова u.

Пример 1.15. Очевидно, $\varepsilon \sqsubset baa$, $b \sqsubset baa$, $ba \sqsubset baa$ и $baa \sqsubset baa$.

Определение 1.16. Говорят, что слово x — *суффикс* (*конец*) слова y (обозначение $x \sqsupset y$), если $y = ux$ для некоторого слова u.

Определение 1.17. Говорят, что слово x — *подслово* (substring) слова y, если $y = uxv$ для некоторых слов u и v.

Определение 1.18. Через $|w|_a$ обозначается количество вхождений символа a в слово w.

Пример 1.19. Если $\Sigma = \{a, b, c\}$, то $|baaa|_a = 3$, $|baaa|_b = 1$ и $|baaa|_c = 0$.

Определение 1.20. Если $L \subseteq \Sigma^*$, то L называется *языком* (или *формальным языком*) над алфавитом Σ.

Поскольку каждый язык является множеством, можно рассматривать операции объединения, пересечения и разности языков, заданных над одним и тем же алфавитом (обозначения $L_1 \cup L_2$, $L_1 \cap L_2$, $L_1 - L_2$).

Пример 1.21. Множество $\{a, abb\}$ является языком над алфавитом $\{a, b\}$.

Пример 1.22. Множество $\{a^k b a^l \mid k \leqslant l\}$ является языком над алфавитом $\{a, b\}$.

Определение 1.23. Пусть $L \subseteq \Sigma^*$. Тогда язык $\Sigma^* - L$ называется *дополнением* (complement) языка L относительно алфавита Σ. Когда из контекста ясно, о каком алфавите идёт речь, говорят просто, что язык $\Sigma^* - L$ является дополнением языка L.

Определение 1.24. Пусть $L_1, L_2 \subseteq \Sigma^*$. Тогда $L_1 \cdot L_2 \rightleftharpoons \{xy \mid x \in L_1,\ y \in L_2\}$. Язык $L_1 \cdot L_2$ называется *конкатенацией* языков L_1 и L_2.

Пример 1.25. Если $L_1 = \{a, abb\}$ и $L_2 = \{bbc, c\}$, то $L_1 \cdot L_2 = \{ac, abbc, abbbbc\}$.

Определение 1.26. Пусть $L \subseteq \Sigma^*$. Тогда $L^0 \rightleftharpoons \{\varepsilon\}$ и $L^n \rightleftharpoons \underbrace{L \cdot \ldots \cdot L}_{n \text{ раз}}$.

Пример 1.27. Если $L = \{a^k b a^l \mid 0 < k < l\}$, то $L^2 = \{a^k b a^l b a^m \mid 0 < k < l - 1,\ m > 1\}$.

Определение 1.28. *Итерацией* (Kleene closure) языка L (обозначение L^*) называется язык $\bigcup\limits_{n \in \mathbb{N}} L^n$. Эта операция называется также *звёздочкой Клини* (Kleene star, star operation).

Пример 1.29. Если $\Sigma = \{a, b\}$ и $L = \{aa, ab, ba, bb\}$, то $L^* = \{w \in \Sigma^* \mid |w| \text{ делится на } 2\}$.

Определение 1.30. *Обращением* или *зеркальным образом* (reversal) слова w (обозначается w^{R}) называется слово, составленное из символов слова w в обратном порядке.

Пример 1.31. Если $w = baaca$, то $w^{\mathrm{R}} = acaab$.

Определение 1.32. Пусть $L \subseteq \Sigma^*$. Тогда $L^{\mathrm{R}} \rightleftharpoons \{w^{\mathrm{R}} \mid w \in L\}$.

1.2. Гомоморфизмы

[Сал, с. 10], [Гин, с. 57], [АхоУль, 0.2.3], [ХопМотУль, 4.2.3, 4.2.4], [Гла, 1.1], [КукБей, с. 259], [LewPap2, с. 85]

Определение 1.33. Пусть Σ_1 и Σ_2 — алфавиты. Если отображение $h\colon \Sigma_1^* \to \Sigma_2^*$ удовлетворяет условию $h(x \cdot y) = h(x) \cdot h(y)$ для всех слов $x \in \Sigma_1^*$ и $y \in \Sigma_1^*$, то отображение h называется *гомоморфизмом* (*морфизмом*).

Замечание 1.34. Можно доказать, что если h — гомоморфизм, то $h(\varepsilon) = \varepsilon$.

Пример 1.35. Пусть $\Sigma_1 = \{a, b\}$ и $\Sigma_2 = \{c\}$. Тогда отображение $h\colon \Sigma_1^* \to \Sigma_2^*$, заданное равенством $h(w) = c^{2|w|}$, является гомоморфизмом.

Замечание 1.36. Каждый гомоморфизм однозначно определяется своими значениями на однобуквенных словах.

Определение 1.37. Если $h\colon \Sigma_1^* \to \Sigma_2^*$ — гомоморфизм и $L \subseteq \Sigma_1^*$, то через $h(L)$ обозначается язык $\{h(w) \mid w \in L\}$.

Пример 1.38. Пусть $\Sigma = \{a, b\}$ и гомоморфизм $h\colon \Sigma^* \to \Sigma^*$ задан равенствами $h(a) = abba$ и $h(b) = \varepsilon$. Тогда $h(\{baa, bb\}) = \{abbaabba, \varepsilon\}$.

Определение 1.39. Если $h\colon \Sigma_1^* \to \Sigma_2^*$ — гомоморфизм и $L \subseteq \Sigma_2^*$, то через $h^{-1}(L)$ обозначается язык $\{w \in \Sigma_1^* \mid h(w) \in L\}$.

Пример 1.40. Рассмотрим алфавит $\Sigma = \{a, b\}$. Пусть гомоморфизм $h\colon \Sigma^* \to \Sigma^*$ задан равенствами $h(a) = ab$ и $h(b) = abb$. Тогда $h^{-1}(\{\varepsilon, abbb, abbab, ababab\}) = \{\varepsilon, ba, aaa\}$.

1.3. Порождающие грамматики

[Гин, 1.1], [Сал, 2.1], [АхоУль, 2.1.2], [Гла, 1.2], [Лал, с. 159–161], [Бра, с. 32–36], [ГлаМел, с. 34–48], [ГорМол, с. 354–355, 367–370], [СокКушБад, с. 12–13], [ТраБар, 1.12], [LewPap2, 4.6], [Рей, с. 28–30], [КукБей, с. 264–268]

Определение 1.41. *Порождающей грамматикой* (*грамматикой типа* 0) (generative grammar, rewrite grammar) называется четвёрка $G \rightleftharpoons \langle N, \Sigma, P, S\rangle$, где N и Σ — конечные алфавиты, $N \cap \Sigma = \varnothing$, $P \subset (N \cup \Sigma)^+ \times (N \cup \Sigma)^*$, P конечно и $S \in N$. Здесь Σ — *основной алфавит* (*терминальный алфавит*), его элементы называются *терминальными символами* или *терминалами* (terminal), N — *вспомогательный алфавит* (*нетерминальный алфавит*), его элементы называются *нетерминальными символами*, *нетерминалами* или *переменными* (nonterminal, variable), S — *начальный символ* (*аксиома*) (start symbol). Пары $(\alpha, \beta) \in P$ называются *правилами подстановки*, просто *правилами* или *продукциями* (rewriting rule, production) и записываются в виде $\alpha \to \beta$.

Пример 1.42. Пусть даны множества $N = \{S\}$, $\Sigma = \{a, b, c\}$, $P = \{S \to acSbcS,\ cS \to \varepsilon\}$. Тогда $\langle N, \Sigma, P, S\rangle$ является порождающей грамматикой.

Замечание 1.43. Будем обозначать элементы множества Σ строчными буквами из начала латинского алфавита, а элементы множества N — заглавными латинскими буквами. Обычно в примерах мы будем задавать грамматику в виде списка правил, подразумевая, что алфавит N составляют все заглавные буквы, встречающиеся в правилах, а алфавит Σ — все строчные буквы, встречающиеся в правилах. При этом правила порождающей грамматики записывают в таком порядке, что левая часть первого правила есть начальный символ S.

Замечание 1.44. Для обозначения n правил с одинаковыми левыми частями $\alpha \to \beta_1, \ \ldots, \ \alpha \to \beta_n$ часто используют сокращённую запись $\alpha \to \beta_1 \mid \ldots \mid \beta_n$.

Определение 1.45. Пусть дана грамматика G. Пишем $\phi \underset{G}{\Rightarrow} \psi$, если $\phi = \eta\alpha\theta$, $\psi = \eta\beta\theta$ и $(\alpha \to \beta) \in P$ для некоторых слов α, β, η, θ в алфавите $N \cup \Sigma$.

Замечание 1.46. Когда из контекста ясно, о какой грамматике идёт речь, вместо $\underset{G}{\Rightarrow}$ можно писать просто $\Rightarrow$.

Пример 1.47. Пусть

$$G = \langle\{S\}, \{a, b, c\}, \{S \to acSbcS, cS \to \varepsilon\}, S\rangle.$$

Тогда $cSacS \underset{G}{\Rightarrow} cSa$.

Определение 1.48. Если $\omega_0 \underset{G}{\Rightarrow} \omega_1 \underset{G}{\Rightarrow} \ldots \underset{G}{\Rightarrow} \omega_n$, где $n \geqslant 0$, то пишем $\omega_0 \overset{*}{\underset{G}{\Rightarrow}} \omega_n$ (другими словами, бинарное отношение $\overset{*}{\underset{G}{\Rightarrow}}$ является рефлексивным, транзитивным замыканием бинарного отношения $\underset{G}{\Rightarrow}$, определённого на множестве $(N \cup \Sigma)^*$). При этом последовательность слов ω_0, ω_1, ..., ω_n называется *выводом* (derivation) слова ω_n из слова ω_0 в грамматике G. Число n называется *длиной* (*количеством шагов*) этого вывода.

Замечание 1.49. В частности, для всякого слова $\omega \in (N \cup \Sigma)^*$ имеет место $\omega \overset{*}{\underset{G}{\Rightarrow}} \omega$ (так как возможен вывод длины 0).

Пример 1.50. Пусть $G = \langle \{S\}, \{a, b\}, \{S \to aSa,\ S \to b\}, S\rangle$. Тогда $aSa \overset{*}{\underset{G}{\Rightarrow}} aaaaSaaaa$. Длина этого вывода — 3.

Определение 1.51. Язык, *порождаемый* грамматикой G, — это множество $L(G) \rightleftharpoons \{\omega \in \Sigma^* \mid S \overset{*}{\underset{G}{\Rightarrow}} \omega\}$. Будем также говорить, что грамматика G *порождаёт* (generates) язык $L(G)$.

Замечание 1.52. Существенно, что в определение порождающей грамматики включены два алфавита — Σ и N. Это позволило нам в определении 1.51 “отсеять” часть слов, получаемых из начального символа. А именно, отбрасывается каждое слово, содержащее хотя бы один символ, не принадлежащий алфавиту Σ.

Пример 1.53. Если $G = \langle \{S\}, \{a, b\}, \{S \to aSa,\ S \to bb\}, S\rangle$, то $L(G) = \{a^n bba^n \mid n \geqslant 0\}$.

Определение 1.54. Две грамматики *эквивалентны*, если они порождают один и тот же язык.

Пример 1.55. Грамматика $S \to abS$, $S \to a$ и грамматика $T \to aU$, $U \to baU$, $U \to \varepsilon$ эквивалентны.

1.4. Классы грамматик

[Гин, с. 23–24, 78–79], [АхоУль, 2.1.3, с. 191], [Сал, 2.1, с. 94], [Гла, 1.2, 1.3], [Бра, с. 39–45], [ГлаМел, с. 54, 63, 69–70], [ГорМол, с. 361–367], [ТраБар, 1.12], [КукБей, с. 268–271], [ЛПИИ, 5.2.1]

Определение 1.56. *Контекстной грамматикой* (*контекстно-зависимой грамматикой, грамматикой непосредственно составляющих, НС-грамматикой, грамматикой типа* 1) (context-sensitive grammar, phrase-structure grammar) называется порождающая грамматика, каждое правило которой имеет вид $\eta A\theta \to \eta\alpha\theta$, где $A \in N$, $\eta \in (N \cup \Sigma)^*$, $\theta \in (N \cup \Sigma)^*$, $\alpha \in (N \cup \Sigma)^+$.

Пример 1.57. Грамматика $S \to TS$, $S \to US$, $S \to b$, $Tb \to Ab$, $A \to a$, $TA \to AAT$, $UAb \to b$, $UAAA \to AAU$ не является контекстной (последние три правила не имеют требуемого вида).

Определение 1.58. *Контекстно-свободной грамматикой* (*КС-грамматикой, бесконтекстной грамматикой, грамматикой типа* 2) (context-free grammar) называется порождающая грамматика, каждое правило которой имеет вид $A \to \alpha$, где $A \in N$, $\alpha \in (N \cup \Sigma)^*$.

Пример 1.59. Грамматика $S \to ASTA$, $S \to AbA$, $A \to a$, $bT \to bb$, $AT \to UT$, $UT \to UV$, $UV \to TV$, $TV \to TA$ является контекстной, но не контекстно-свободной (последние пять правил не имеют требуемого вида).

Определение 1.60. *Линейной грамматикой* (linear grammar) называется порождающая грамматика, каждое правило которой имеет вид $A \to u$ или $A \to uBv$, где $A \in N$, $u \in \Sigma^*$, $v \in \Sigma^*$, $B \in N$.

Пример 1.61. Грамматика $S \to TT$, $T \to cTT$, $T \to bT$, $T \to a$ является контекстно-свободной, но не линейной (первые два правила не имеют требуемого вида).

Определение 1.62. *Праволинейной грамматикой* (*рациональной грамматикой, грамматикой типа* 3) (right-linear grammar) называется порождающая грамматика, каждое правило которой имеет вид $A \to u$ или $A \to uB$, где $A \in N$, $u \in \Sigma^*$, $B \in N$.

Пример 1.63. Грамматика $S \to aSa$, $S \to T$, $T \to bT$, $T \to \varepsilon$ является линейной, но не праволинейной (первое правило не имеет требуемого вида).

Пример 1.64. Грамматика $S \to T$, $U \to abba$ праволинейная.

Пример 1.65. Грамматика $S \to aS$, $S \to bS$, $S \to aaaT$, $S \to aabaT$, $S \to abaaT$, $S \to aabbaT$, $S \to ababaT$, $S \to abbaaT$, $T \to aT$, $T \to bT$, $T \to \varepsilon$ праволинейная.

Пример 1.66. Грамматика $S \to \varepsilon$, $S \to aaaS$, $S \to abbS$, $S \to babS$, $S \to aabT$, $T \to abaT$, $T \to baaT$, $T \to bbbT$, $T \to bbaS$ праволинейная. Обобщённый вариант языка, порождаемого этой грамматикой, используется в доказательстве разрешимости арифметики Пресбургера [Sip, с. 207–208].

Определение 1.67. Правила вида $\alpha \to \varepsilon$ называются *ε-правилами*.

Лемма 1.68. *Каждая праволинейная грамматика является линейной. Каждая линейная грамматика является контекстно-свободной. Каждая контекстно-свободная грамматика без ε-правил является контекстной грамматикой.*

Определение 1.69. Классы грамматик типа 0, 1, 2 и 3 образуют *иерархию Хомского* (Chomsky hierarchy).

Определение 1.70. Язык называется *контекстным языком* (*контекстно-свободным языком*, *линейным языком*, *праволинейным языком*), если он порождается некоторой контекстной грамматикой (соответственно контекстно-свободной грамматикой, линейной грамматикой, праволинейной грамматикой). Контекстно-свободные языки называются также *алгебраическими языками*.

Пример 1.71. Пусть дан произвольный алфавит $\Sigma = \{a_1, \dots a_n\}$. Тогда язык Σ^* является праволинейным, так как он порождается грамматикой $S \to \varepsilon$, $S \to a_1 S$, ..., $S \to a_n S$.

2. Конечные автоматы

2.1. Недетерминированные конечные автоматы

[Гин, 2.1], [Сал, 2.1], [АхоУль, 2.2.3], [ХопМотУль, 2.3], [Гла, 5.1], [Лал, с. 185], [ГорМол, с. 391–392], [СокКушБад, с. 19–21], [ЛПИИ, 5.2.3], [Бра, с. 106–107], [ГроЛан, 10.3.1], [ТраБар, 1.5], [LewPap2, 2.2], [Sip, 1.2], [Рей, с. 46–47], [КукБей, с. 324–325], [АхоСетУль, 3.6]

Определение 2.1. *Конечный автомат* (finite automaton, finite-state machine) — это пятёрка $M = \langle Q, \Sigma, \Delta, I, F \rangle$, где Σ — конечный алфавит, Q и Δ — конечные множества, $\Delta \subseteq Q \times \Sigma^* \times Q$, $I \subseteq Q$, $F \subseteq Q$. Элементы Q называются *состояниями* (state), элементы I — *начальными* (initial) состояниями, элементы F — *заключительными* или *допускающими* (final, accepting) состояниями. Если $\langle p, x, q \rangle \in \Delta$, то $\langle p, x, q \rangle$ называется *переходом* (transition) из p в q, а слово x — *меткой* (label) этого перехода.

Пример 2.2. Пусть $Q = \{1, 2\}$, $\Sigma = \{a, b\}$, $I = \{2\}$, $F = \{2\}$, $\Delta = \{\langle 1, aaa, 1\rangle,\ \langle 1, ab, 2\rangle,\ \langle 1, b, 2\rangle,\ \langle 2, \varepsilon, 1\rangle\}$. Тогда $\langle Q, \Sigma, \Delta, I, F\rangle$ — конечный автомат.

Определение 2.3. *Путь* (path) конечного автомата — это кортеж $\langle q_0, e_1, q_1, e_2, \ldots, q_n\rangle$, где $n \geqslant 0$ и $e_i = \langle q_{i-1}, w_i, q_i\rangle \in \Delta$ для каждого i. При этом q_0 — *начало пути*, q_n — *конец пути*, n — *длина пути*, $w_1 \ldots w_n$ — *метка пути*.

Замечание 2.4. Для любого состояния $q \in Q$ существует путь $\langle q\rangle$. Его метка — ε, начало и конец совпадают.

Определение 2.5. Путь называется *успешным*, если его начало принадлежит I, а конец принадлежит F.

Пример 2.6. Рассмотрим конечный автомат из примера 2.2. Путь $\langle 2,\ \langle 2, \varepsilon, 1\rangle,\ 1,\ \langle 1, aaa, 1\rangle,\ 1,\ \langle 1, b, 2\rangle,\ 2\rangle$ является успешным. Его метка — $aaab$. Длина этого пути — 3.

Определение 2.7. Слово w *допускается* (is accepted, is recognized) конечным автоматом M, если оно является меткой некоторого успешного пути.

Определение 2.8. *Язык, распознаваемый конечным автоматом M,* — это язык $L(M)$, состоящий из меток всех успешных путей (то есть из всех допускаемых данным автоматом слов). Будем также говорить, что автомат M *распознаёт* (accepts) язык $L(M)$.

Замечание 2.9. Если $I \cap F \neq \varnothing$, то язык, распознаваемый конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, содержит ε.

Пример 2.10. Пусть $M = \langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2\}$, $\Sigma = \{a, b\}$, $\Delta = \{\langle 1, a, 2\rangle,\ \langle 2, b, 1\rangle\}$, $I = \{1\}$ и $F = \{1, 2\}$. Тогда $L(M) = \{(ab)^n \mid n \geqslant 0\} \cup \{(ab)^n a \mid n \geqslant 0\}$.

Определение 2.11. Два конечных автомата *эквивалентны*, если они распознают один и тот же язык.

Определение 2.12. Язык L называется *автоматным* (finite-state language), если существует конечный автомат, распознающий этот язык.

Замечание 2.13. Обычно в учебниках используется более узкое определение недетерминированного конечного автомата, но это не меняет класса автоматных языков (см. лемму 2.30).

Пример 2.14. Рассмотрим однобуквенный алфавит $\Sigma = \{a\}$. При любых фиксированных $k \in \mathbb{N}$ и $m \in \mathbb{N}$ язык $\{a^{k+mn} \mid n \in \mathbb{N}\}$ является автоматным.

Замечание 2.15. Каждый конечный язык является автоматным.

Определение 2.16. Состояние q *достижимо* (reachable) из состояния p, если существует путь, началом которого является p, а концом — q.

Лемма 2.17. *Каждый автоматный язык распознаётся некоторым конечным автоматом, в котором каждое состояние достижимо из некоторого начального состояния и из каждого состояния достижимо хотя бы одно заключительное состояние.*

Пример 2.18. Рассмотрим конечный автомат $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2, 3, 4\}$, $\Sigma = \{a, b, c, d\}$, $I = \{1, 2, 4\}$, $F = \{1, 3, 4\}$, $\Delta = \{\langle 1, a, 2\rangle,\ \langle 1, b, 4\rangle,\ \langle 3, c, 4\rangle,\ \langle 4, d, 1\rangle\}$. Удалим те состояния (и содержащие их переходы), которые не удовлетворяют требованиям леммы 2.17. Получается эквивалентный конечный автомат $\langle Q', \Sigma, \Delta', I', F'\rangle$, где $Q' = \{1, 4\}$, $\Delta' = \{\langle 1, b, 4\rangle,\ \langle 4, d, 1\rangle\}$, $I' = \{1, 4\}$ и $F' = \{1, 4\}$.

Замечание 2.19. Естественным обобщением конечного автомата является *конечный преобразователь* (finite-state transducer), позволяющий на каждом такте отправить несколько символов в дополнительный “выходной” поток (см. [Гин, 3.3], [АхоУль, 3.1.3] или [ТраБар, 0.1, 2.1–2.6]). В некоторых книгах конечными автоматами называют именно такие устройства (с выходным потоком).

В данном пособии конечные преобразователи рассматриваться не будут.

2.2*. Конфигурации конечного автомата

[АхоУль, 2.2.3], [ГорМол, с. 391], [СокКушБад, с. 20], [LewPap2, 2.2]

Определение 2.20. *Конфигурацией* конечного автомата называется любая упорядоченная пара $\langle q, w\rangle$, где $q \in Q$ и $w \in \Sigma^*$.

Замечание 2.21. Содержательно конфигурация представляет собой “мгновенное описание” конечного автомата. Если представить, что исходное слово, принадлежность которого рассматриваемому языку надо проверить, дано в некотором “входном потоке”, то в конфигурации $\langle q, w\rangle$ слово w есть та часть исходного слова, которая пока осталась во входном потоке (это некоторый суффикс исходного слова), а q — текущее состояние "управляющего устройства".

Определение 2.22. Определим на множестве всех конфигураций конечного автомата M бинарное отношение $\vdash$ (*такт работы* (step)) следующим образом. Если $\langle p, x, q\rangle \in \Delta$ и $w \in \Sigma^*$, то $\langle p, xw\rangle \vdash \langle q, w\rangle$. Иногда вместо $\vdash$ пишут $\underset{\Delta}{\vdash}$.

Пример 2.23. Рассмотрим конечный автомат из примера 2.2. Тогда $\langle 1, abba\rangle \vdash \langle 2, ba\rangle$.

Определение 2.24. Бинарное отношение $\overset{*}{\vdash}$ определяется как рефлексивное, транзитивное замыкание отношения $\vdash$.

Пример 2.25. Для конечного автомата из примера 2.2 выполняется $\langle 1, aaaab\rangle \overset{*}{\vdash} \langle 1, aaaab\rangle$ и $\langle 1, aaaab\rangle \overset{*}{\vdash} \langle 2, \varepsilon\rangle$.

Лемма 2.26. *Пусть дан конечный автомат* $M = \langle Q, \Sigma, \Delta, I, F\rangle$. *Слово* $w \in \Sigma^*$ *принадлежит языку* $L(M)$ *тогда и только тогда, когда для некоторых* $p \in I$ *и* $q \in F$ *верно* $\langle p, w\rangle \overset{*}{\vdash} \langle q, \varepsilon\rangle$.

Лемма 2.27. *Если* $\langle q_1, x\rangle \overset{*}{\vdash} \langle q_2, \varepsilon\rangle$ *и* $\langle q_2, y\rangle \overset{*}{\vdash} \langle q_3, \varepsilon\rangle$, *то* $\langle q_1, xy\rangle \overset{*}{\vdash} \langle q_3, \varepsilon\rangle$.

2.3. Конечные автоматы с однобуквенными переходами

[ХопМотУль, 2.5], [Рей, с. 51–53]

Лемма 2.28. *Каждый автоматный язык распознаётся некоторым конечным автоматом, не содержащим переходов с метками длины больше единицы и имеющим ровно одно начальное состояние и ровно одно заключительное состояние.*

Пример 2.29. Рассмотрим язык, заданный конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2\}$, $\Sigma = \{a, b, c, d\}$, $\Delta = \{\langle 1, ab, 2\rangle, \langle 2, cd, 1\rangle\}$, $I = \{1, 2\}$, $F = \{1, 2\}$. Тот же язык распознаётся конечным автоматом $\langle Q', \Sigma, \Delta', I', F'\rangle$, где $Q' = \{0, 1, 2, 3, 4, 5\}$, $I' = \{0\}$, $F' = \{5\}$, $\Delta' = \{\langle 1, a, 3\rangle, \langle 3, b, 2\rangle, \langle 2, c, 4\rangle, \langle 4, d, 1\rangle, \langle 0, \varepsilon, 1\rangle, \langle 0, \varepsilon, 2\rangle, \langle 1, \varepsilon, 5\rangle, \langle 2, \varepsilon, 5\rangle\}$. Здесь первые два перехода заменяют старый переход $\langle 1, ab, 2\rangle$ и следующие два перехода заменяют старый переход $\langle 2, cd, 1\rangle$. Чтобы обеспечить единственность начального состояния, добавлены переходы $\langle 0, \varepsilon, 1\rangle$ и $\langle 0, \varepsilon, 2\rangle$. Последние два перехода в Δ' обеспечивают единственность заключительного состояния.

Лемма 2.30. *Каждый автоматный язык распознаётся некоторым конечным автоматом, содержащим только переходы с метками длины единица и имеющим ровно одно начальное состояние.*

Доказательство. Согласно лемме 2.28 можно предположить, что исходный язык задан конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, не содержащим переходов с метками длины больше единицы, причём $|I| = 1$. Построим искомый конечный автомат $\langle Q', \Sigma, \Delta', I', F'\rangle$, положив $Q' = Q$, $I' = I$,

$$\Delta' = \{\langle p, a, r\rangle \mid a \in \Sigma \text{ и найдётся такое } q \in Q, \text{ что } \langle q, a, r\rangle \in \Delta \\ \text{и существует путь из } p \text{ в } q \text{ с меткой } \varepsilon\},$$

$$F' = \{p \in Q \mid \text{найдётся такое } q \in F, \\ \text{что существует путь из } p \text{ в } q \text{ с меткой } \varepsilon\}.$$

□

2.4. Характеризация праволинейных языков

[Гин, 2.2], [Сал, 2.1], [ХопМотУль, с. 196], [Гла, 5.1], [Лал, с. 186–187], [ЛПИИ, 5.2.3, 5.2.4], [АхоУль, 2.2.4], [ГорМол, с. 408–413], [Бра, с. 107–108], [ГроЛан, 10.2.4], [LewPap1, 3.2], [Рей, с. 48–50], [КукБей, с. 340–342], [АхоСетУль, с. 180]

Теорема 2.31. *Каждый автоматный язык является праволинейным.*

Доказательство. Без ограничения общности можно предположить, что исходный язык задан конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q \cap \Sigma = \varnothing$ и $I = \{q_0\}$. Положим $N = Q$, $S = q_0$ и $P = \{p \to xq \mid \langle p, x, q\rangle \in \Delta\} \cup \{p \to \varepsilon \mid p \in F\}$. □

Пример 2.32. Язык, распознаваемый конечным автоматом из примера 2.2, порождается грамматикой $K_2 \to K_1$, $K_1 \to aaaK_1$, $K_1 \to abK_2$, $K_1 \to bK_2$, $K_2 \to \varepsilon$.

Теорема 2.33. *Каждый праволинейный язык является автоматным.*

Доказательство. Без ограничения общности можно предположить, что исходный язык задан праволинейной грамматикой, не содержащей правил вида $A \to u$, где $u \in \Sigma^+$. Положим $Q = N$, $I = \{S\}$, $F = \{A \in N \mid (A \to \varepsilon) \in P\}$ и $\Delta = \{\langle A, u, B\rangle \mid (A \to uB) \in P\}$. □

Пример 2.34. Пусть $\Sigma = \{a, b\}$. Язык, порождаемый грамматикой $S \to aa$, $S \to T$, $T \to baT$, $T \to a$, распознаётся конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{S, T, E\}$, $I = \{S\}$, $F = \{E\}$ и $\Delta = \{\langle S, aa, E\rangle,\ \langle S, \varepsilon, T\rangle,\ \langle T, ba, T\rangle,\ \langle T, a, E\rangle\}$.

2.5*. Нормальная форма праволинейных грамматик

[АхоУль, с. 145], [ГорМол, с. 387–390], [Бра, с. 77–78]

Определение 2.35. *Праволинейная грамматика в нормальной форме* (*автоматная грамматика, регулярная грамматика*) (finite-state grammar) — это праволинейная грамматика, в которой каждое правило имеет вид $A \to \varepsilon$, $A \to a$ или $A \to aB$, где $A \in N$, $B \in N$, $a \in \Sigma$.

Теорема 2.36. *Каждая праволинейная грамматика эквивалентна некоторой праволинейной грамматике в нормальной форме.*

Теорема 2.37. *Если праволинейный язык не содержит пустого слова, то он порождается некоторой праволинейной грамматикой в нормальной форме без ε-правил.*

2.6. Детерминированные конечные автоматы

[Гин, 2.1], [Сал, 2.2], [АхоУль, 2.2.3], [ХопМотУль, 2.2], [Гла, 5.1], [ГорМол, с. 392–395], [СокКушБад, с. 21], [Лал, с. 174–178, 185], [Бра, с. 101–102], [ГроЛан, 10.2], [ТраБар, 0.1, 0.2, 1.7], [LewPap2, 2.1], [Sip, 1.1, 1.2], [Рей, с. 44–48], [КукБей, с. 320–327], [АхоСетУль, 3.6]

Определение 2.38. Конечный автомат $\langle Q, \Sigma, \Delta, I, F\rangle$ называется *детерминированным* (deterministic), если

1) множество I содержит ровно один элемент,
2) для каждого перехода $\langle p, x, q\rangle \in \Delta$ выполняется равенство $|x| = 1$,

3) для любого символа $a \in \Sigma$ и для любого состояния $p \in Q$ существует не более одного состояния $q \in Q$ со свойством $\langle p, a, q\rangle \in \Delta$.

Пример 2.39. Конечный автомат из примера 2.10 является детерминированным.

Определение 2.40. Детерминированный конечный автомат $\langle Q, \Sigma, \Delta, I, F\rangle$ называется *полным* (complete), если для каждого состояния $p \in Q$ и для каждого символа $a \in \Sigma$ найдётся такое состояние $q \in Q$, что $\langle p, a, q\rangle \in \Delta$.

Пример 2.41. Конечный автомат из примера 2.10 эквивалентен полному детерминированному конечному автомату $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2, 3\}$, $\Sigma = \{a, b\}$, $\Delta = \{\langle 1, a, 2\rangle,\ \langle 1, b, 3\rangle,\ \langle 2, a, 3\rangle,\ \langle 2, b, 1\rangle,\ \langle 3, a, 3\rangle,\ \langle 3, b, 3\rangle\}$, $I = \{1\}$, $F = \{1, 2\}$.

Замечание 2.42. Некоторые авторы используют в определении полного детерминированного конечного автомата вместо отношения Δ функцию $\delta\colon Q \times \Sigma \to Q$. От функции δ можно перейти к отношению Δ, положив $\Delta = \{\langle p, a, \delta(p, a)\rangle \mid p \in Q,\ a \in \Sigma\}$.

Теорема 2.43. *Каждый автоматный язык распознаётся некоторым полным детерминированным конечным автоматом.*

Доказательство. Без ограничения общности можно предположить, что исходный язык задан конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, содержащим только переходы с метками длины единица. Для любых $a \in \Sigma$ и $H \subseteq Q$ обозначим $\Delta_a(H) \rightleftharpoons \{q \in Q \mid \langle p, a, q\rangle \in \Delta \text{ для некоторого } p \in H\}$.

Обозначим через $\mathcal{P}(Q)$ множество всех подмножеств множества Q. Построим искомый полный детерминированный конечный автомат $\langle Q', \Sigma, \Delta', I', F'\rangle$, положив $Q' = \mathcal{P}(Q)$, $\Delta' = \{\langle H, a, \Delta_a(H)\rangle \mid H \subseteq Q,\ a \in \Sigma\}$, $I' = \{I\}$ и $F' = \{H \subseteq Q \mid H \cap F \neq \varnothing\}$. $\square$

Пример 2.44. Пусть $\Sigma = \{a, b\}$. Рассмотрим конечный автомат $M = \langle\{1, 2, 3\}, \Sigma, \Delta, \{1\}, \{3\}\rangle$, где $\Delta = \{\langle 1, a, 1\rangle,\ \langle 1, b, 1\rangle,\ \langle 1, a, 2\rangle,\ \langle 2, b, 3\rangle\}$. Если применить конструкцию из доказательства теоремы 2.43 и потом удалить состояния, не достижимые из начального состояния, то получится полный детерминированный конечный автомат $M' = \langle\{\{1\},\{1,2\},\{1,3\}\},\Sigma,\Delta',\{\{1\}\},\{\{1,3\}\}\rangle$, где

$$\Delta' = \{\langle\{1\},a,\{1,2\}\rangle,\ \langle\{1\},b,\{1\}\rangle,\ \langle\{1,2\},a,\{1,2\}\rangle,\\ \langle\{1,2\},b,\{1,3\}\rangle,\ \langle\{1,3\},a,\{1,2\}\rangle,\ \langle\{1,3\},b,\{1\}\rangle\}.$$

Определение 2.45. Для любого состояния p полного детерминированного конечного автомата $\langle Q,\Sigma,\Delta,I,F\rangle$ и любого слова w обозначим через $\Delta_w(p)$ такое состояние q, что существует путь из p в q с меткой w (в силу полноты и детерминированности такое состояние существует и единственно).

3. Основные свойства автоматных языков

3.1. Свойства замкнутости класса автоматных языков

[Гин, 2.1], [Сал, 2.3, с. 58], [ХопМотУль, 3.2.3, 4.2.2, 4.2.5], [АхоУль, 2.3.3], [Гла, 5.2], [ГроЛан, 10.4.1, 10.4.3], [ТраБар, 1.1, 1.6], [LewPap2, 2.3], [Sip, 1.2], [Рей, с. 40–43], [КукБей, с. 327–330], [АхоСетУль, 3.7], [ГорМол, с. 413]

Теорема 3.1. *Класс автоматных языков замкнут относительно итерации, конкатенации и объединения.*

Доказательство. Без ограничения общности можно предположить, что каждый из исходных языков задан конечным автоматом с одним начальным и одним заключительным состоянием. Тогда во всех трёх случаях результирующий автомат получается из исходных путём добавления нескольких ε-переходов и состояний и назначения новых начальных и заключительных состояний. □

Пример 3.2. Пусть $M_1 = \langle\{1,2,3\},\Sigma,\Delta_1,\{1\},\{2\}\rangle$, где $\Sigma = \{a,b,c\}$ и $\Delta_1 = \{\langle 1,a,2\rangle,\ \langle 2,b,3\rangle,\ \langle 3,c,1\rangle\}$. Тогда язык $L(M_1)^*$ распознаётся конечным автоматом $M_2 = \langle\{1,2,3,4\},\Sigma,\Delta_1\cup\{\langle 4,\varepsilon,1\rangle,\ \langle 2,\varepsilon,4\rangle\},\{4\},\{4\}\rangle$.

Пример 3.3. Пусть $\Sigma = \{a, b, c\}$. Рассмотрим конечный автомат M_1 из примера 3.2 и конечный автомат $M_2 = \langle\{4,5\}, \Sigma, \Delta_2, \{4\}, \{5\}\rangle$, где $\Delta_2 = \{\langle 4, c, 4\rangle,\ \langle 4, a, 5\rangle,\ \langle 5, c, 5\rangle\}$. Тогда язык $L(M_1) \cdot L(M_2)$ распознаётся конечным автоматом $M_3 = \langle\{1,2,3,4,5\}, \Sigma, \Delta_1 \cup \Delta_2 \cup \{\langle 2, \varepsilon, 4\rangle\}, \{1\}, \{5\}\rangle$, а язык $L(M_1) \cup L(M_2)$ распознаётся конечным автоматом $M_4 = \langle\{1,2,3,4,5\}, \Sigma, \Delta_1 \cup \Delta_2, \{1,4\}, \{2,5\}\rangle$.

Упражнение 3.4. Существует ли такой автоматный язык L, что язык L^{R} не является автоматным?
Ответ. Нет. Пусть язык L распознаётся конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$. Положим $\Delta^{\mathrm{R}} = \{\langle q, x^{\mathrm{R}}, p\rangle \mid \langle p, x, q\rangle \in \Delta\}$. Тогда язык L^{R} распознаётся конечным автоматом $\langle Q, \Sigma, \Delta^{\mathrm{R}}, F, I\rangle$.

Упражнение 3.5. Существует ли такой автоматный язык $L \subseteq \Sigma^*$, что язык

$$\mathrm{Pref}(L) \rightleftharpoons \{w \mid w \text{ — префикс некоторого слова из } L\}$$

не является автоматным?
Ответ. Нет. Пусть язык L распознаётся конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, причём

1) в Δ нет переходов с метками длины больше единицы и
2) из каждого состояния достижимо некоторое заключительное состояние.

Легко проверить, что язык $\mathrm{Pref}(L)$ распознаётся конечным автоматом $\langle Q, \Sigma, \Delta, I, Q\rangle$.

Упражнение 3.6. Существует ли такой автоматный язык $L \subseteq \Sigma^*$, что язык

$$\mathrm{Suf}(L) \rightleftharpoons \{w \mid w \text{ — суффикс некоторого слова из } L\}$$

не является автоматным?
Ответ. Нет. $\mathrm{Suf}(L) = \mathrm{Pref}(L^{\mathrm{R}})^{\mathrm{R}}$.

Упражнение 3.7. Существует ли такой автоматный язык $L \subseteq \Sigma^*$, что язык

$$\mathrm{Subw}(L) \rightleftharpoons \{w \mid w \text{ — подслово некоторого слова из } L\}$$

не является автоматным?
Ответ. Нет. $\mathrm{Subw}(L) = \mathrm{Suf}(\mathrm{Pref}(L))$.

3.2. Пересечение и дополнение автоматных языков

[Гин, 2.1], [ХопМотУль, 4.2.1], [АхоУль, 2.3.3], [Сал, 2.2], [Гла, 5.2], [ГроЛан, 10.4.2], [ТраБар, 1.1, 1.6], [LewPap2, 2.3], [Рей, с. 50–51], [КукБей, с. 329], [ГорМол, с. 413]

Теорема 3.8. *Класс автоматных языков замкнут относительно дополнения и пересечения.*

Доказательство. Если язык L распознаётся полным детерминированным конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, то язык $\Sigma^* - L$ распознаётся конечным автоматом $\langle Q, \Sigma, \Delta, I, Q - F\rangle$.

Пересечение выражается через объединение и дополнение (закон де Моргана). □

Замечание 3.9. Автоматность пересечения двух автоматных языков можно легко доказать и без привлечения теоремы 2.43. Для этого достаточно построить по двум конечным автоматам с однобуквенными переходами $M_1 = \langle Q_1, \Sigma, \Delta_1, I_1, F_1\rangle$ и $M_2 = \langle Q_2, \Sigma, \Delta_2, I_2, F_2\rangle$ новый конечный автомат $M = \langle Q_1 \times Q_2, \Sigma, \Delta, I_1 \times I_2, F_1 \times F_2\rangle$, где $\Delta = \{\langle\langle p_1, p_2\rangle, a, \langle q_1, q_2\rangle\rangle \mid \langle p_1, a, q_1\rangle \in \Delta_1,\ \langle p_2, a, q_2\rangle \in \Delta_2\}$.

3.3. Лемма о разрастании для автоматных языков

[АхоУль, 2.3.2], [ХопМотУль, 4.1], [ГорМол, с. 414–415], [LewPap2, 2.4], [Sip, 1.4], [Сал, с. 37], [Рей, с. 62]

Лемма 3.10 (pumping lemma, лемма о разрастании, лемма о накачке, лемма-насос). *Пусть L — автоматный язык над алфавитом Σ. Тогда найдётся такое положительное целое число p, что для любого слова $w \in L$ длины не меньше p найдутся слова $x, y, z \in \Sigma^*$, для которых верно $xyz = w$, $y \neq \varepsilon$, $|xy| \leqslant p$ и $xy^iz \in L$ для всех $i \geqslant 0$.*

Доказательство. Пусть язык L распознаётся автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, содержащим только переходы с метками длины единица. Положим $p = |Q|$. Пусть слово w является меткой успешного пути $\langle q_0, e_1, q_1, e_2, \ldots, q_n\rangle$ и $|w| = n \geqslant p$. Согласно принципу Дирихле найдутся такие индексы j и k, что

$0 \leqslant j < k \leqslant p$ и $q_j = q_k$. Выберем слова x, y и z так, что $|x| = j$, $|y| = k - j$ и $xyz = w$. □

Пример 3.11. Пусть $\Sigma = \{a, b\}$. Рассмотрим автоматный язык $L = \{(ab)^n \mid n \geqslant 0\} \cup \{a(ab)^n \mid n \geqslant 0\}$. Положим $p = 3$. Тогда для любого слова $w \in L$ длины не меньше p найдутся слова $x, y, z \in \Sigma^*$, соответствующие утверждению леммы 3.10. Действительно, если $w = abu$ для некоторого слова u, то положим $x = \varepsilon$, $y = ab$, $z = u$; иначе $w = aabu$ и можно положить $x = a$, $y = ab$, $z = u$.

3.4. Примеры неавтоматных языков

[АхоУль, 2.3.2], [ХопМотУль, 4.1], [ГорМол, с. 415], [LewPap2, 2.4]

Пример 3.12. Рассмотрим язык $L = \{ab^n a^n \mid n \geqslant 0\}$ над алфавитом $\Sigma = \{a, b\}$. Утверждение леммы 3.10 не выполняется ни для какого натурального числа p. Действительно, если $w = ab^p a^p$, то $x = ab^k$, $y = b^m$, $z = b^{p-k-m} a^p$ для некоторых $k \geqslant 0$ и $m \geqslant 1$ или $x = \varepsilon$, $y = ab^l$, $z = b^{p-l} a^p$ для некоторого $l \geqslant 0$. В обоих случаях $xyyz \notin L$. Таким образом, язык L не является автоматным.

Упражнение 3.13. Пусть $\Sigma = \{a, b, c\}$. При каких словах $u \in \{a, b\}^*$ и $v \in \{a, b\}^*$ язык $\{u^m c v^m \mid m > 0\}$ является автоматным?

Ответ. Этот язык является автоматным тогда и только тогда, когда $u = \varepsilon$ или $v = \varepsilon$.

Замечание 3.14. Условие, сформулированное в лемме 3.10, является необходимым для автоматности, но не достаточным.

Пример 3.15. Пусть $\Sigma = \{a, b\}$. Рассмотрим язык $L = \{a^k b^m a^n \mid k = 0 \ \text{ или } \ m = n\}$. Положим $p = 1$. Тогда для любого слова $w \in L$ длины не меньше p найдутся слова $x, y, z \in \Sigma^*$, соответствующие утверждению леммы 3.10. Тем не менее язык L не является автоматным, так как $L \cap \{ab^m a^n \mid m \geqslant 0,\ n \geqslant 0\} = \{ab^n a^n \mid n \geqslant 0\}$.

4. Дополнительные свойства автоматных языков

4.1. Гомоморфизмы и автоматные языки

[АхоУль, с. 161], [ХопМотУль, 4.2.3, 4.2.4], [LewPap2, с. 85–86]

Теорема 4.1. *Для любого гомоморфизма $h\colon \Sigma_1^* \to \Sigma_2^*$ и автоматного языка $L \subseteq \Sigma_1^*$ язык $h(L)$ является автоматным.*

Доказательство. Пусть исходный язык L задан конечным автоматом $M = \langle Q, \Sigma_1, \Delta, I, F\rangle$. Положим $\Delta' = \{\langle p, h(x), q\rangle \mid \langle p, x, q\rangle \in \Delta\}$. Тогда язык $h(L)$ распознаётся конечным автоматом $\langle Q, \Sigma_2, \Delta', I, F\rangle$. □

Теорема 4.2. *Для любого гомоморфизма $h\colon \Sigma_1^* \to \Sigma_2^*$ и автоматного языка $L \subseteq \Sigma_2^*$ язык $h^{-1}(L)$ является автоматным.*

Доказательство. Без ограничения общности можно предположить, что исходный язык L задан конечным автоматом $M = \langle Q, \Sigma_2, \Delta, I, F\rangle$, где Δ не содержит переходов с метками длины больше единицы. Положим $\Delta' = \{\langle p, a, q\rangle \mid a \in \Sigma_1$ и существует путь из p в q с меткой $h(a)\}$. Тогда язык $h^{-1}(L)$ распознаётся конечным автоматом $M' = \langle Q, \Sigma_1, \Delta', I, F\rangle$. □

4.2*. Длины слов в автоматных языках

[Гин, 2.1], [АхоУль, с. 147], [Сал, с. 40]

Определение 4.3. Пусть $\mathcal{A} \subseteq \mathbb{N}$, $m \in \mathbb{N}$ и $m > 0$. Множество $\mathcal{A}$ называется *заключительно периодическим* (ultimately periodic) с периодом m, если выполнено условие $(\exists n_0 \in \mathbb{N})\,(\forall n \geqslant n_0)\,(n \in \mathcal{A} \leftrightarrow n + m \in \mathcal{A})$.

Лемма 4.4. *Пусть $\mathcal{A} \subseteq \mathbb{N}$. Тогда равносильны следующие утверждения:*

1) *множество $\mathcal{A}$ является заключительно периодическим;*
2) *найдутся положительное целое число m и конечные множества $\mathcal{M} \subseteq \mathbb{N}$ и $\mathcal{K} \subseteq \{0, 1, \ldots, m-1\}$, такие что $\mathcal{A} = \{k \in \mathbb{N} \mid (k \bmod m) \in \mathcal{K}\} - \mathcal{M}$;*
3) *множество $\mathcal{A}$ является объединением конечного семейства арифметических прогрессий.*

Теорема 4.5. *Язык L над однобуквенным алфавитом $\{a\}$ является автоматным тогда и только тогда, когда множество $\{k \in \mathbb{N} \mid a^k \in L\}$ является заключительно периодическим.*

Доказательство. Для доказательства необходимости достаточно рассмотреть детерминированный конечный автомат, распознающий язык L. □

Теорема 4.6. *Если язык L является автоматным, то множество $\{|w| \mid w \in L\}$ является заключительно периодическим.*

Доказательство. Рассмотрим конечный автомат, распознающий язык L. Заменим все символы в метках переходов на символ a. Осталось применить теорему 4.5 к полученному автоматному языку над однобуквенным алфавитом $\{a\}$. □

Упражнение 4.7. Существует ли такой автоматный язык L над алфавитом $\{a\}$, что язык $\{a^n \mid a^{n^2} \in L\}$ не является автоматным?

Ответ. Нет. Пусть L — автоматный язык. Согласно теореме 4.5 найдутся положительное целое число m и конечные множества $\mathcal{M} \subseteq \mathbb{N}$ и $\mathcal{K} \subseteq \{0, 1, \ldots, m-1\}$, такие что $\{k \in \mathbb{N} \mid a^k \in L\} = \{k \in \mathbb{N} \mid (k \bmod m) \in \mathcal{K}\} - \mathcal{M}$. Положим $\mathcal{M}' = \{n \in \mathbb{N} \mid n^2 \in \mathcal{M}\}$ и $\mathcal{K}' = \{n \in \mathbb{N} \mid n < m,\ (n^2 \bmod m) \in \mathcal{K}\}$. Тогда $\{n \in \mathbb{N} \mid a^{n^2} \in L\} = \{n \in \mathbb{N} \mid (n \bmod m) \in \mathcal{K}'\} - \mathcal{M}'$.

Упражнение 4.8. Существует ли такой автоматный язык L над алфавитом $\{a\}$, что язык $\{a^n \mid a^{2^n} \in L\}$ не является автоматным?

Ответ. Нет. Пусть L — автоматный язык. Согласно теореме 4.5 найдутся положительное целое число m и конечные множества $\mathcal{M} \subseteq \mathbb{N}$ и $\mathcal{K} \subseteq \{0, 1, \ldots, m-1\}$, такие что $\{k \in \mathbb{N} \mid a^k \in L\} = \{k \in \mathbb{N} \mid (k \bmod m) \in \mathcal{K}\} - \mathcal{M}$. Осталось проверить, что язык $\{a^n \mid (2^n \bmod m) \in \mathcal{K}\}$ распознаётся конечным автоматом $\langle Q, \{a\}, \Delta, I, F\rangle$, где $Q = \{0, 1, \ldots, m-1\}$, $\Delta = \{\langle k, a, (2k \bmod m)\rangle \mid k \in Q\}$, $I = \{2^0 \bmod m\}$ и $F = \mathcal{K}$.

Упражнение 4.9. Существует ли такой автоматный язык L_1 над алфавитом Σ, что язык

$$L_2 = \{w \in \Sigma^* \mid wx \in L_1,\ |x| = 2^{|w|} \text{ для некоторого слова } x \in \Sigma^*\}$$

не является автоматным?

Ответ. Нет. Пусть язык L_1 распознаётся конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, не содержащим переходов с метками длины больше единицы. Обозначим $M_q = \langle Q, \Sigma, \Delta, I, \{q\}\rangle$ и $M'_q = \langle Q, \Sigma, \Delta, \{q\}, F\rangle$. Тогда

$$L_2 = \bigcup_{q \in Q} (L(M_q) \cap \mathcal{A}_q),$$

где $\mathcal{A}_q = \{w \in \Sigma^* \mid x \in L(M'_q),\ |x| = 2^{|w|}$ для некоторого $x \in \Sigma^*\}$. Из ответа на предыдущее упражнение следует (по теоремам 4.1 и 4.2), что языки $\mathcal{A}_q$ являются автоматными.

5. Регулярные выражения

5.1. Определение регулярного выражения

[Сал, 2.3], [АхоУль, 2.2.1], [ХопМотУль, 3.1], [Гла, 5.2], [СокКушБад, с. 25–26], [ГорМол, с. 397–398], [LewPap2, 1.8], [Sip, 1.3], [Рей, с. 54–55], [КукБей, с. 335], [АхоСетУль, 3.3]

Определение 5.1. *Регулярное выражение* над алфавитом Σ определяется рекурсивно следующим образом: 0 является регулярным выражением; 1 является регулярным выражением; если $a \in \Sigma$, то a является регулярным выражением; если e и f являются регулярными выражениями, то $(e+f)$, $(e \cdot f)$ и e^* тоже являются регулярными выражениями.

Для экономии скобок будем считать, что умножение связывает теснее, чем сложение. Вместо $e \cdot f$ часто пишут просто ef.

Пример 5.2. Пусть $\Sigma = \{a, b\}$. Тогда $((a \cdot b)^* \cdot (1+a))$ является регулярным выражением над алфавитом Σ.

Определение 5.3. Каждое регулярное выражение e над алфавитом Σ *задаёт* (denotes, represents) некоторый язык над алфавитом Σ (обозначение $L(e)$), определяемое рекурсивно следующим образом:

$$\begin{aligned} L(a) &\rightleftharpoons \{a\}, \text{ если } a \in \Sigma, \\ L(0) &\rightleftharpoons \varnothing, \\ L(1) &\rightleftharpoons \{\varepsilon\}, \\ L(e+f) &\rightleftharpoons L(e) \cup L(f), \end{aligned}$$

$$L(e{\cdot}f) \rightleftharpoons L(e) \cdot L(f),$$
$$L(e^*) \rightleftharpoons L(e)^*.$$

Заметим, что в правой части последнего выражения символом $*$ обозначена итерация языка (см. определение 1.28).

Вместо $L(e)$ часто пишут просто e.

Пример 5.4. Пусть $\Sigma = \{a, b\}$. Согласно определению $L((ab)^*{\cdot}(1{+}a)) = \{(ab)^n \mid n \geqslant 0\} \cup \{(ab)^n a \mid n \geqslant 0\}$.

Определение 5.5. Язык L называется *регулярным*, если он задаётся некоторым регулярным выражением.

Определение 5.6. Пусть e — регулярное выражение. Тогда $e^+ \rightleftharpoons e^*e$.

5.2*. Свойства регулярных выражений

[АхоУль, 2.2.1], [ХопМотУль, 3.4], [Сал, с. 37], [СокКушБад, с. 27–28], [ГорМол, с. 398], [Рей, с. 55–56], [КукБей, с. 335–336]

Лемма 5.7. *Регулярные выражения образуют ассоциативное полукольцо с операциями* $(0, +, 1, \cdot)$*, то есть для любых регулярных выражений* e*,* f *и* g *выполняются следующие тождества:*

1) $e{+}f = f{+}e$,
2) $e{+}0 = e$,
3) $(e{+}f){+}g = e{+}(f{+}g)$,
4) $e{\cdot}1 = e$,
5) $1{\cdot}e = e$,
6) $(e{\cdot}f){\cdot}g = e{\cdot}(f{\cdot}g)$,
7) $e{\cdot}(f{+}g) = e{\cdot}f{+}e{\cdot}g$,
8) $(f{+}g){\cdot}e = f{\cdot}e{+}g{\cdot}e$,
9) $e{\cdot}0 = 0$,
10) $0{\cdot}e = 0$.

Равенство понимается как равенство языков, задаваемых регулярными выражениями.

Лемма 5.8. *Для любых регулярных выражений* e *и* f *выполняются следующие тождества:*

1) $e{+}e = e$,
2) $(1{+}e{+}ee{+}\ldots{+}e^{n-1})(e^n)^* = e^*$ *для любого* $n \geqslant 1$,
3) $(e^*f)^*e^* = (e{+}f)^*$,

4) $1+e(fe)^*f = (ef)^*$.

Лемма 5.9. *Для любых регулярных выражений e, f и g, если $e = ef+g$ и $\varepsilon \notin L(f)$, то $e = gf^*$.*

Упражнение 5.10. Упростить регулярное выражение 0^*.
Ответ. $0^* = 1$.

5.3. Теорема Клини

[Сал, 2.4], [ХопМотУль, 3.2], [Гла, 5.2], [ГроЛан, 10.4.4], [ГорМол, с. 404–408], [LewPap2, 2.3], [Рей, с. 56–57]

Определение 5.11. Назовём *обобщённым конечным автоматом* аналог конечного автомата, где переходы помечены не словами, а регулярными выражениями. *Метка пути* такого автомата — произведение регулярных выражений на переходах данного пути. Слово *w допускается* обобщённым конечным автоматом, если оно принадлежит языку, задаваемому меткой некоторого успешного пути.

Замечание 5.12. Каждый конечный автомат можно преобразовать в обобщённый конечный автомат, допускающий те же слова. Для этого достаточно заменить всюду в метках переходов пустое слово на 1, а каждое непустое слово на произведение его букв.

Замечание 5.13. Если к обобщённому конечному автомату добавить переход с меткой 0, то множество допускаемых этим автоматом слов не изменится.

Пример 5.14. Пусть $\Sigma = \{a, b, c\}$. Обобщённый конечный автомат $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2, 3\}$, $\Delta = \{\langle 1, a, 2\rangle, \ \langle 2, b^*ba, 2\rangle, \ \langle 2, b^*, 3\rangle\}$, $I = \{1, 2\}$ и $F = \{3\}$, допускает все слова в алфавите Σ, кроме слов, содержащих подслово aa.

Теорема 5.15 (теорема Клини). *Язык L является регулярным тогда и только тогда, когда он является автоматным.*

Доказательство. Пусть e — регулярное выражение. Индукцией по построению e легко показать, что задаваемый им язык является автоматным (см. теорему 3.1).

Обратно, пусть язык L распознаётся некоторым (недетерминированным) конечным автоматом с одним начальным состоянием и одним заключительным состоянием. Существует эквивалентный ему обобщённый конечный автомат $\langle Q, \Sigma, \Delta, \{q_1\}, \{q_2\}\rangle$, где $q_1 \neq q_2$. Если есть несколько переходов с общим началом и общим концом (такие переходы называются *параллельными*), заменим их на один переход, используя операцию $+$.

Устраним по очереди все состояния, кроме q_1 и q_2. При устранении состояния q надо для каждого перехода вида $\langle p_1, f_1, q\rangle$, где $p_1 \neq q$, и для каждого перехода вида $\langle q, f_2, p_2\rangle$, где $p_2 \neq q$, добавить переход $\langle p_1, f_1 g^* f_2, p_2\rangle$, где регулярное выражение g — метка перехода из q в q (если такого перехода нет, то считаем, что $g = 0$), и снова всюду заменить параллельные переходы на один переход, используя операцию $+$.

После устранения всех состояний, кроме q_1 и q_2, получится обобщённый конечный автомат $\langle \{q_1, q_2\}, \Sigma, \Delta', \{q_1\}, \{q_2\}\rangle$, где $\Delta' = \{\langle q_1, e_{11}, q_1\rangle, \langle q_1, e_{12}, q_2\rangle, \langle q_2, e_{21}, q_1\rangle, \langle q_2, e_{22}, q_2\rangle\}$. Очевидно, $L = L(e_{11}^* e_{12}(e_{22}+e_{21}e_{11}^*e_{12})^*)$. □

Пример 5.16. Рассмотрим язык, распознаваемый конечным автоматом $M = \langle \{q_1, q_2, q_3, q_4\}, \Sigma, \Delta, \{q_1\}, \{q_2\}\rangle$, где $\Sigma = \{a, b, c\}$ и

$$\begin{aligned}\Delta = \{&\langle q_1, a, q_4\rangle,\ \langle q_2, cb, q_2\rangle,\ \langle q_2, bac, q_4\rangle,\\ &\langle q_3, a, q_1\rangle,\ \langle q_3, c, q_2\rangle,\ \langle q_3, c, q_4\rangle,\ \langle q_4, \varepsilon, q_3\rangle,\ \langle q_4, b, q_4\rangle\}.\end{aligned}$$

Тот же язык порождается обобщённым конечным автоматом $M_1 = \langle \{q_1, q_2, q_3, q_4\}, \Sigma, \Delta_1, \{q_1\}, \{q_2\}\rangle$, где

$$\begin{aligned}\Delta_1 = \{&\langle q_1, a, q_4\rangle,\ \langle q_2, cb, q_2\rangle,\ \langle q_2, bac, q_4\rangle,\\ &\langle q_3, a, q_1\rangle,\ \langle q_3, c, q_2\rangle,\ \langle q_3, c, q_4\rangle,\ \langle q_4, 1, q_3\rangle,\ \langle q_4, b, q_4\rangle\}.\end{aligned}$$

После устранения состояния q_3 получается обобщённый конечный автомат $M_2 = \langle \{q_1, q_2, q_4\}, \Sigma, \Delta_2, \{q_1\}, \{q_2\}\rangle$, где

$$\begin{aligned}\Delta_2 = \{&\langle q_1, a, q_4\rangle,\ \langle q_2, cb, q_2\rangle,\ \langle q_2, bac, q_4\rangle,\\ &\langle q_4, 1{\cdot}0^*{\cdot}a, q_1\rangle,\ \langle q_4, 1{\cdot}0^*{\cdot}c, q_2\rangle,\ \langle q_4, b+1{\cdot}0^*{\cdot}c, q_4\rangle\}.\end{aligned}$$

Можно упростить регулярные выражения и получить

$$\begin{aligned}\Delta_2' = \{&\langle q_1, a, q_4\rangle,\ \langle q_2, cb, q_2\rangle,\ \langle q_2, bac, q_4\rangle,\\ &\langle q_4, a, q_1\rangle,\ \langle q_4, c, q_2\rangle,\ \langle q_4, b+c, q_4\rangle\}.\end{aligned}$$

После устранения состояния q_4 и упрощения регулярных выражений получается обобщённый конечный автомат $M_3 = \langle \{q_1, q_2\}, \Sigma, \Delta_3, \{q_1\}, \{q_2\} \rangle$, где

$$\Delta_3 = \{ \langle q_1, a(b+c)^*a, q_1 \rangle,\ \langle q_1, a(b+c)^*c, q_2 \rangle,\\ \langle q_2, bac(b+c)^*a, q_1 \rangle,\ \langle q_2, cb+bac(b+c)^*c, q_2 \rangle \}.$$

Следовательно, язык $L(M)$ задаётся регулярным выражением

$$(a(b+c)^*a)^*a(b+c)^*c\cdot\\ \cdot(cb+bac(b+c)^*c+bac(b+c)^*a(a(b+c)^*a)^*a(b+c)^*c)^*.$$

Упростив это регулярное выражение, получим

$$L(M) = L(a(aa+b+c+c(cb)^*bac)^*c(cb)^*).$$

6. Синтаксические моноиды

6.1. Множества правых контекстов

[АхоУль, с. 157–158], [Лал, с. 179–180], [LewPap2, 2.5], [ТраБар, 1.2], [ГроЛан, 13.2.3, 14.3.1–14.3.5], [Рей, с. 57–59], [Сал, с. 37]

Определение 6.1. Пусть $L \subseteq \Sigma^*$ и $y \in \Sigma^*$. Тогда *множество правых контекстов* слова y относительно языка L определяется так: $\mathrm{C}_L^{(\mathrm{r})}(y) \rightleftharpoons \{ z \in \Sigma^* \mid yz \in L \}$.

Пример 6.2. Пусть $\Sigma = \{a, b\}$ и $L = \{a^n b a^n \mid n \geqslant 0\}$. Тогда

1) $\mathrm{C}_L^{(\mathrm{r})}(a^i) = \{a^k b a^{k+i} \mid k \geqslant 0\}$,

2) если $i \geqslant j$, то $\mathrm{C}_L^{(\mathrm{r})}(a^i b a^j) = \{a^{i-j}\}$,

3) если $i < j$, то $\mathrm{C}_L^{(\mathrm{r})}(a^i b a^j) = \varnothing$,

4) если $|y|_b > 1$, то $\mathrm{C}_L^{(\mathrm{r})}(y) = \varnothing$.

Лемма 6.3. *Если язык L распознаётся полным детерминированным конечным автоматом $\langle Q, \Sigma, \Delta, I, F \rangle$, то $|\{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\}| \leqslant |Q|$.*

Доказательство. Пусть $I = \{s\}$. Введём обозначение $J \rightleftharpoons \{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\}$. Определим функцию $f\colon J \to Q$, положив $f(A)$ равным $\Delta_y(s)$, где y — некоторое слово, для которого выполнено условие $\mathrm{C}_L^{(\mathrm{r})}(y) = A$ (если существует несколько таких слов y, то можно использовать, например, первое среди них в лексикографическом порядке).

Заметим, что для любых слов u и v, если $\mathrm{C}_L^{(\mathrm{r})}(u) \neq \mathrm{C}_L^{(\mathrm{r})}(v)$, то $\Delta_u(s) \neq \Delta_v(s)$. Следовательно, функция f является инъективной. Но тогда $|J| \leqslant |Q|$. □

Пример 6.4. Рассмотрим язык L, порождаемый полным детерминированным конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$ из примера 2.41. Тогда $\{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\} = \{\mathrm{C}_L^{(\mathrm{r})}(\varepsilon), \mathrm{C}_L^{(\mathrm{r})}(a), \mathrm{C}_L^{(\mathrm{r})}(b)\}$, $\mathrm{C}_L^{(\mathrm{r})}(\varepsilon) = \{(ab)^n \mid n \geqslant 0\} \cup \{(ab)^n a \mid n \geqslant 0\}$, $\mathrm{C}_L^{(\mathrm{r})}(a) = \{b(ab)^n \mid n \geqslant 0\} \cup \{(ba)^n \mid n \geqslant 0\}$ и $\mathrm{C}_L^{(\mathrm{r})}(b) = \varnothing$.

Лемма 6.5. *Если $L \subseteq \Sigma^*$ и множество $\{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\}$ конечно, то язык L является автоматным.*

Доказательство. Язык L распознаётся полным детерминированным конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\}$, $I = \{\mathrm{C}_L^{(\mathrm{r})}(\varepsilon)\}$, $F = \{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in L\}$, $\Delta = \{\langle \mathrm{C}_L^{(\mathrm{r})}(y), a, \mathrm{C}_L^{(\mathrm{r})}(ya)\rangle \mid y \in \Sigma^*,\ a \in \Sigma\}$. □

Пример 6.6. Пусть $\Sigma = \{a, b\}$. Рассмотрим автоматный язык $L = a^+b^*$. Обозначим $q_\varepsilon = \mathrm{C}_L^{(\mathrm{r})}(\varepsilon)$, $q_a = \mathrm{C}_L^{(\mathrm{r})}(a)$, $q_b = \mathrm{C}_L^{(\mathrm{r})}(b) = \varnothing$, $q_{ab} = \mathrm{C}_L^{(\mathrm{r})}(ab)$. Тогда $\{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\} = \{q_\varepsilon, q_a, q_b, q_{ab}\}$. Язык L распознаётся полным детерминированным конечным автоматом $\langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{q_\varepsilon, q_a, q_b, q_{ab}\}$, $I = \{q_\varepsilon\}$, $F = \{q_a, q_{ab}\}$,

$$\begin{aligned}\Delta = \{&\langle q_\varepsilon, a, q_a\rangle,\ \langle q_\varepsilon, b, q_b\rangle,\ \langle q_a, a, q_a\rangle,\ \langle q_a, b, q_{ab}\rangle,\\ &\langle q_b, a, q_b\rangle,\ \langle q_b, b, q_b\rangle,\ \langle q_{ab}, a, q_b\rangle,\ \langle q_{ab}, b, q_{ab}\rangle\}.\end{aligned}$$

Теорема 6.7. *Язык $L \subseteq \Sigma^*$ является автоматным тогда и только тогда, когда множество $\{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\}$ конечно.*

Доказательство. Необходимость доказана в лемме 6.3, достаточность — в лемме 6.5. □

Замечание 6.8. В силу леммы 6.3 полный детерминированный конечный автомат, построенный в доказательстве леммы 6.5, является минимальным (по количеству состояний) среди всех полных детерминированных конечных автоматов, распознающих заданный язык. Можно доказать, что любой минимальный полный детерминированный конечный автомат, распознающий заданный язык, изоморфен этому автомату.

6.2. Минимизация детерминированных конечных автоматов

[АхоУль, 2.3.1], [Сал, с. 36–37], [ХопМотУль, 4.4], [ГорМол, с. 395–397], [Рей, с. 59–61], [LewPap2, 2.5]

Определение 6.9. Говорят, что слово w *различает* (distinguishes) состояния p и q полного детерминированного конечного автомата $\langle Q, \Sigma, \Delta, I, F\rangle$, если одно из состояний $\Delta_w(p)$, $\Delta_w(q)$ заключительное, а другое не является заключительным.

Определение 6.10. Состояния p и q полного детерминированного конечного автомата $\langle Q, \Sigma, \Delta, I, F\rangle$ называются *различимыми* (distinguishable), если существует слово w, которое их различает.

Замечание 6.11. Пусть полный детерминированный конечный автомат $\langle Q, \Sigma, \Delta, \{q_{\mathrm{s}}\}, F\rangle$ задаёт язык L. Рассмотрим произвольные слова $u \in \Sigma^*$ и $v \in \Sigma^*$. Состояния $\Delta_u(q_{\mathrm{s}})$ и $\Delta_v(q_{\mathrm{s}})$ различимы тогда и только тогда, когда $\mathrm{C}_L^{(\mathrm{r})}(u) \neq \mathrm{C}_L^{(\mathrm{r})}(v)$.

Теорема 6.12. *Существует быстрый алгоритм, позволяющий по произвольному детерминированному конечному автомату находить минимальный (по количеству состояний) автомат среди детерминированных конечных автоматов, эквивалентных исходному автомату.*

Доказательство. Без ограничения общности можно предположить, что дан полный детерминированный конечный автомат $\langle Q, \Sigma, \Delta, I, F\rangle$, все состояния которого достижимы из начального состояния (для обеспечения полноты достаточно добавить одно состояние). Для каждого натурального числа i определим на множестве Q отношение эквивалентности $\equiv_i$:

$$p \equiv_0 q \iff (p \in F \text{ и } q \in F) \text{ или } (p \notin F \text{ и } q \notin F),$$
$$p \equiv_{i+1} q \iff p \equiv_i q \text{ и } \Delta_a(p) \equiv_i \Delta_a(q) \text{ для каждого } a \in \Sigma.$$

Легко видеть, что $p \equiv_i q$ тогда и только тогда, когда никакое слово длины не больше i не различает p и q. Обозначим $n = |Q|$. Легко проверить, что для любого $k \geqslant n$ отношение $\equiv_k$ совпадает с отношением $\equiv_{n-1}$. Используя классы эквивалентности отношения $\equiv_{n-1}$ как состояния, можно построить минимальный полный детерминированный конечный автомат. Удалив из него бесполезное состояние (из которого не достижимо ни одно заключительное состояние), если такое имеется, получим искомый минимальный детерминированный конечный автомат. □

Пример 6.13. Рассмотрим полный детерминированный конечный автомат $\langle\{1,2,3,4,5,6\},\{a,b\},\Delta,\{1\},\{1,3\}\rangle$, где

$$\Delta=\{\langle 1,a,2\rangle,\ \langle 2,b,3\rangle,\ \langle 3,a,4\rangle,\ \langle 4,b,1\rangle,\ \langle 4,a,5\rangle,\ \langle 5,a,5\rangle,\\ \langle 1,b,6\rangle,\ \langle 2,a,6\rangle,\ \langle 3,b,6\rangle,\ \langle 5,b,6\rangle,\ \langle 6,a,6\rangle,\ \langle 6,b,6\rangle\}.$$

Он эквивалентен минимальному детерминированному конечному автомату $\langle\{\{1,3\},\{2,4\}\},\{a,b\},\Delta',\{\{1,3\}\},\{\{1,3\}\}\rangle$, где

$$\Delta'=\{\langle\{1,3\},a,\{2,4\}\rangle,\ \langle\{2,4\},b,\{1,3\}\rangle\}.$$

Замечание 6.14. Неизвестно, существует ли быстрый (полиномиальный) алгоритм, позволяющий по произвольному конечному автомату находить минимальный автомат среди всех (не обязательно детерминированных) конечных автоматов, эквивалентных исходному автомату.

6.3. Множества двусторонних контекстов

[Лал, с. 180], [Гин, 2.1], [Гла, 5.2], [ТраБар, 1.2], [ГроЛан, 13.2.1, 14.3.5], [АхоУль, с. 158]

Определение 6.15. Пусть $L\subseteq\Sigma^*$ и $y\in\Sigma^*$. Тогда *множество контекстов* (*множество двусторонних контекстов*) слова y относительно языка L определяется так: $\mathrm{C}_L(y)\rightleftharpoons\{\langle x,z\rangle\in\Sigma^*\times\Sigma^*\mid xyz\in L\}$.

Пример 6.16. Пусть $\Sigma=\{a,b\}$ и $L=\{a^nba^n\mid n\geqslant 0\}$. Тогда

$$\mathrm{C}_L(a^i)=\{\langle a^m,a^kba^{k+i+m}\rangle\mid m\geqslant 0,\ k\geqslant 0\}\cup\\ \cup\{\langle a^{k+i+m}ba^k,a^m\rangle\mid k\geqslant 0,\ m\geqslant 0\},$$

$$\mathrm{C}_L(a^iba^j)=\{\langle a^k,a^m\rangle\mid k\geqslant 0,\ m\geqslant 0,\ k+i=m+j\},$$

$$\mathrm{C}_L^{(\mathrm{r})}(y)=\varnothing,\ \text{если } |y|_b>1.$$

Лемма 6.17. *Если* $\mathrm{C}_L(u_1)=\mathrm{C}_L(u_2)$, *то* $\mathrm{C}_L^{(\mathrm{r})}(u_1)=\mathrm{C}_L^{(\mathrm{r})}(u_2)$.

Доказательство. Из определений следует, что $\mathrm{C}_L^{(\mathrm{r})}(y)=\{z\in\Sigma^*\mid\langle\varepsilon,z\rangle\in\mathrm{C}_L(y)\}$. □

Лемма 6.18. *Если* $\mathrm{C}_L(u_1) = \mathrm{C}_L(u_2)$, *то* $\mathrm{C}_L(u_1 v) = \mathrm{C}_L(u_2 v)$ *и* $\mathrm{C}_L(v u_1) = \mathrm{C}_L(v u_2)$.

Доказательство. Пусть $\mathrm{C}_L(u_1) = \mathrm{C}_L(u_2)$ и $\langle x, z\rangle \in \mathrm{C}_L(u_1 v)$. Тогда $x u_1 v z \in L$. Следовательно, $\langle x, vz\rangle \in \mathrm{C}_L(u_1)$. Далее, $\langle x, vz\rangle \in \mathrm{C}_L(u_2)$, $x u_2 v z \in L$ и $\langle x, z\rangle \in \mathrm{C}_L(u_2 v)$. Второе равенство доказывается аналогично. □

Лемма 6.19. *Если* $\mathrm{C}_L(u_1) = \mathrm{C}_L(u_2)$ *и* $\mathrm{C}_L(v_1) = \mathrm{C}_L(v_2)$, *то* $\mathrm{C}_L(u_1 v_1) = \mathrm{C}_L(u_2 v_2)$.

Определение 6.20. Пусть $L \subseteq \Sigma^*$. Тогда множество $\mathrm{Synt}(L) \rightleftharpoons \{\mathrm{C}_L(y) \mid y \in \Sigma^*\}$ называется *синтаксическим моноидом* (syntactic monoid) языка L.

Теорема 6.21. *Синтаксический моноид* $\mathrm{Synt}(L)$ *конечен тогда и только тогда, когда язык* L *является автоматным.*

Доказательство. Пусть множество $\mathrm{Synt}(L)$ конечно. Согласно лемме 6.17 множество $\{\mathrm{C}_L^{(\mathrm{r})}(y) \mid y \in \Sigma^*\}$ тоже конечно. В силу леммы 6.5 язык L является автоматным.

Обратно, пусть язык L распознаётся конечным автоматом $M = \langle Q, \Sigma, \Delta, I, F\rangle$, не содержащим переходов с метками длины больше единицы. Поставим каждому слову y в соответствие множество $\mathrm{Tr}_M(y) \subseteq Q \times Q$, определённое так: $\mathrm{Tr}_M(y) = \{\langle p, q\rangle \in Q \times Q \mid$ существует путь из p в q с меткой $y\}$. Легко проверить, что если $\mathrm{Tr}_M(y_1) = \mathrm{Tr}_M(y_2)$, то $\mathrm{C}_L(y_1) = \mathrm{C}_L(y_2)$. Следовательно, $|\mathrm{Synt}(L)| \leqslant 2^{n^2}$, где $n = |Q|$. □

Лемма 6.22. *Пусть* $L \subseteq \Sigma^*$, $n \in \mathbb{N}$ *и для каждого слова* $y \in \Sigma^*$ *длины* n *найдётся такое слово* $x \in \Sigma^*$, *что* $|x| < n$ *и* $\mathrm{C}_L(x) = \mathrm{C}_L(y)$. *Тогда* $\mathrm{Synt}(L) = \{\mathrm{C}_L(y) \mid y \in \Sigma^*,\ |y| < n\}$.

Доказательство. Индукцией по $k \geqslant n$ можно доказать, что для каждого слова $y \in \Sigma^*$ длины k найдётся такое слово $x \in \Sigma^*$, что $|x| < n$ и $\mathrm{C}_L(x) = \mathrm{C}_L(y)$. В шаге индукции используется лемма 6.19. □

7. Неоднозначность в контекстно-свободных грамматиках

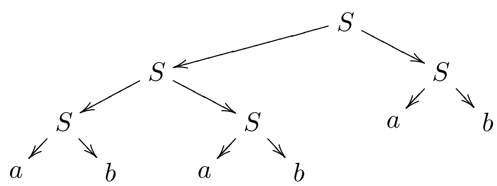
7.1. Деревья вывода

[Гин, 1.5], [ХопМотУль, 5.2.1, 5.2.2], [СокКушБад, с. 13–14, 31], [Бра, с. 45–56], [Лал, с. 294], [АхоУль, 0.5.1–0.5.4, 2.4.1], [ГорМол, с. 370–372], [ГроЛан, 8.2.1], [Гла, 3.1], [LewPap2, 3.2], [Рей, с. 31–32], [АхоСетУль, 2.2, 4.2], [КукБей, с. 277]

Определение 7.1. Выводам в контекстно-свободной грамматике соответствуют так называемые *деревья вывода* (или *деревья разбора*) (derivation tree, parse tree) — некоторые упорядоченные деревья, вершины которых помечены символами алфавита $N \cup \Sigma$. Корень дерева отвечает начальному символу. Каждому символу слова w_1, на которую заменяется начальный символ на первом шаге вывода, ставится в соответствие вершина дерева, и к ней проводится дуга из корня. Полученные таким образом непосредственные потомки корня упорядочены согласно порядку их меток в слове w_1. Для тех из полученных вершин, которые помечены символами из множества N, делается аналогичное построение и т. д.

Кроной (yield) дерева вывода называется слово, записанное в вершинах, помеченных символами из алфавита Σ.

Пример 7.2. Рассмотрим контекстно-свободную грамматику $S \to SS$, $S \to ab$, $S \to aSb$. Выводу $S \Rightarrow SS \Rightarrow Sab \Rightarrow SSab \Rightarrow$ $\Rightarrow abSab \Rightarrow ababab$ соответствует следующее дерево вывода.



7.2. Однозначные контекстно-свободные грамматики

[Гин, 1.6], [АхоУль, 2.4.1, 2.6.5], [ХопМотУль, 5.1.4, 5.2.3–5.2.6, 5.4], [Бра, с. 56–58], [Лал, с. 307], [ГорМол, с. 372–374], [СокКушБад, с. 14], [ГроЛан, 8.2.3], [Гла, 3.1, 4.3], [LewPap2, 3.2], [Sip, 2.1], [Рей, с. 32], [АхоСетУль, 2.2, 4.2], [КукБей, с. 272–275]

Определение 7.3. Вывод в контекстно-свободной грамматике называется *левосторонним* или *левым* (leftmost derivation), если на каждом шаге вывода заменяется самое левое из всех вхождений вспомогательных символов (то есть каждый шаг вывода имеет вид $uA\theta \Rightarrow u\beta\theta$, где $(A \to \beta) \in P$, $u \in \Sigma^*$ и $\theta \in (N \cup \Sigma)^*$). Иногда в левосторонних выводах вместо $\Rightarrow$ пишут $\underset{\mathrm{lm}}{\Rightarrow}$. *Правосторонний вывод* определяется аналогично.

Пример 7.4. Вывод $S \Rightarrow SS \Rightarrow Sab \Rightarrow SSab \Rightarrow abSab \Rightarrow$ $\Rightarrow ababab$ из примера 7.2 не является левосторонним.

Лемма 7.5. *Для каждого слова, выводимого в контекстно-свободной грамматике, существует левосторонний вывод.*

Лемма 7.6. *Пусть G — контекстно-свободная грамматика над алфавитом Σ. Пусть $w \in \Sigma^*$. Тогда существует взаимно-однозначное соответствие между левосторонними выводами слова w в грамматике G и деревьями вывода в грамматике G, кроной которых является w.*

Пример 7.7. Рассмотрим дерево вывода из примера 7.2. Ему соответствует левосторонний вывод $S \underset{\mathrm{lm}}{\Rightarrow} SS \underset{\mathrm{lm}}{\Rightarrow} SSS \underset{\mathrm{lm}}{\Rightarrow} abSS \underset{\mathrm{lm}}{\Rightarrow}$ $\underset{\mathrm{lm}}{\Rightarrow} ababS \underset{\mathrm{lm}}{\Rightarrow} ababab$.

Определение 7.8. Контекстно-свободная грамматика называется *неоднозначной* (ambiguous), если существует слово, которое имеет два или более левосторонних вывода. (Устаревший термин — *неопределённая* грамматика.) В противном случае контекстно-свободная грамматика называется *однозначной* (unambiguous).

Пример 7.9. Контекстно-свободная грамматика из примера 7.2 неоднозначна. Слово $ababab$ имеет два левосторонних вывода: $S \underset{\mathrm{lm}}{\Rightarrow} SS \underset{\mathrm{lm}}{\Rightarrow} SSS \underset{\mathrm{lm}}{\Rightarrow} abSS \underset{\mathrm{lm}}{\Rightarrow} ababS \underset{\mathrm{lm}}{\Rightarrow} ababab$ и $S \underset{\mathrm{lm}}{\Rightarrow} SS \underset{\mathrm{lm}}{\Rightarrow}$ $\underset{\mathrm{lm}}{\Rightarrow} abS \underset{\mathrm{lm}}{\Rightarrow} abSS \underset{\mathrm{lm}}{\Rightarrow} ababS \underset{\mathrm{lm}}{\Rightarrow} ababab$.

Пример 7.10. Пусть $\Sigma = \{\boldsymbol{p}, \#,), (, \neg, \wedge, \vee\}$. Контекстно-свободная грамматика $S \to S{\vee}D$, $S \to D$, $D \to D{\wedge}C$, $D \to C$, $C \to \neg C$, $C \to (S)$, $C \to V$, $V \to V\#$, $V \to \boldsymbol{p}$ порождает множество всех булевых формул, составленных из переменных $\boldsymbol{p}$, $\boldsymbol{p}\#$, $\boldsymbol{p}\#\#$, $\ldots$ с помощью скобок и операторов $\neg$, $\wedge$ и $\vee$. Эта грамматика является однозначной.

Определение 7.11. Контекстно-свободный язык называется *существенно неоднозначным* (inherently ambiguous), если каждая контекстно-свободная грамматика, порождающая этот язык, является неоднозначной.

Пример 7.12. Язык, порождаемый контекстно-свободной грамматикой из примера 7.2, не является существенно неоднозначным. Он порождается однозначной грамматикой $S \to TS$, $S \to T$, $T \to ab$, $T \to aSb$.

Пример 7.13*. Пусть $\Sigma = \{a, b, c\}$. Контекстно-свободный язык $\{a^k b^m c^n \mid k = m \text{ или } m = n\}$ является существенно неоднозначным. Доказательство этого факта можно найти в [АхоУль, с. 234–236].

7.3*. Однозначные праволинейные грамматики

[Лал, с. 187], [АхоУль, с. 119]

Теорема 7.14. *Каждый праволинейный язык порождается некоторой однозначной праволинейной грамматикой. Другими словами, ни один праволинейный язык не является существенно неоднозначным.*

Доказательство. Согласно теоремам 2.33 и 2.43 исходный язык распознаётся некоторым детерминированным конечным автоматом. Применив к нему конструкцию из доказательства теоремы 2.31, получим однозначную праволинейную грамматику. □

7.4. Языки Дика и Лукасевича

[Сал, с. 103, 116], [Лал, с. 293–295], [ЛПИИ, 5.1.3], [АхоУль, с. 239], [Гла, 6.5], [ГроЛан, 15.1]

Определение 7.15. *Языком Дика* (Dyck language) над $2n$ буквами называется контекстно-свободный язык над алфавитом $\{a_1, b_1, a_2, b_2, \ldots, a_n, b_n\}$, порождаемый грамматикой $S \to \varepsilon$, $S \to a_1 S b_1 S$, $\ldots$, $S \to a_n S b_n S$.

Замечание 7.16. Словами этого языка являются последовательности правильно вложенных скобок n типов.

Замечание 7.17. При любом положительном целом n грамматика из определения 7.15 является однозначной.

Определение 7.18. *Языком Лукасевича* (Lukasiewicz language) над $n+1$ буквами называется контекстно-свободный язык над алфавитом $\{a_0, a_1, \ldots, a_n\}$, порождаемый грамматикой $S \to a_0$, $S \to a_1 S$, $S \to a_2 SS$, $\ldots$, $S \to a_n S^n$.

Замечание 7.19. При любом $n \in \mathbb{N}$ грамматика из определения 7.18 является однозначной.

8. Нормальные формы контекстно-свободных грамматик

8.1. Устранение бесполезных символов

[АхоУль, 2.4.2], [ХопМотУль, 7.1.1, 7.1.2], [ГорМол, с. 427–429], [Гла, 4.2], [Бра, с. 61–64], [Рей, с. 65–66], [КукБей, с. 285–287]

Определение 8.1. Пусть дана порождающая грамматика $G = \langle N, \Sigma, P, S \rangle$. Символ $A \in N$ называется *полезным* (useful), если существуют такие слова $\alpha \in (N \cup \Sigma)^*$, $\beta \in (N \cup \Sigma)^*$ и $w \in \Sigma^*$, что $S \overset{*}{\Rightarrow} \alpha A \beta$ и $\alpha A \beta \overset{*}{\Rightarrow} w$. Символ $A \in N$ называется *бесполезным* (useless), если он не является полезным. Символ $A \in N$ называется *порождающим* (generating), если существует такое слово $w \in \Sigma^*$, что $A \overset{*}{\Rightarrow} w$. Символ $A \in N$ называется *достижимым* (reachable), если существуют такие слова $\alpha \in (N \cup \Sigma)^*$ и $\beta \in (N \cup \Sigma)^*$, что $S \overset{*}{\Rightarrow} \alpha A \beta$.

Теорема 8.2. *Пусть дана контекстно-свободная грамматика $G = \langle N, \Sigma, P, S\rangle$ и $L(G) \neq \varnothing$. Тогда существуют такие множества $N' \subseteq N$ и $P' \subseteq P$, что в контекстно-свободной грамматике $\langle N', \Sigma, P', S\rangle$ нет бесполезных символов и она эквивалентна исходной грамматике.*

Доказательство. Сначала удалим все непорождающие символы (удалим также каждое правило, содержащее хотя бы один такой символ). Затем из полученной грамматики удалим все недостижимые символы (и правила, их содержащие). □

Пример 8.3. Рассмотрим контекстно-свободную грамматику G с правилами $S \to UX$, $S \to VZ$, $T \to aa$, $T \to bb$, $U \to aUa$, $U \to bUb$, $V \to aTb$, $V \to bTa$, $W \to YZY$, $W \to aab$, $X \to Xa$, $X \to Xb$, $X \to \varepsilon$, $Y \to YY$, $Y \to aU$, $Y \to \varepsilon$, $Z \to W$, $Z \to b$. Удалив четыре правила, содержащие непорождающий символ U, получим грамматику G_1. В ней символ X является недостижимым. Удалив три правила, содержащие X, получим грамматику G_2 с правилами $S \to VZ$, $T \to aa$, $T \to bb$, $V \to aTb$, $V \to bTa$, $W \to YZY$, $W \to aab$, $Y \to YY$, $Y \to \varepsilon$, $Z \to W$, $Z \to b$. Очевидно, $L(G) = L(G_2)$ и грамматика G_2 не содержит бесполезных символов.

8.2. Устранение ε-правил

[Гин, 1.8], [АхоУль, 2.4.2], [ХопМотУль, 7.1.3], [Лал, с. 295–296], [ГорМол, с. 429–431], [Гла, 4.2], [Бра, с. 71–76], [ГроЛан, 7.2.6], [Сал, 6.2], [Sip, 2.1], [Рей, с. 35–37], [КукБей, с. 288–289]

Теорема 8.4. *Пусть язык L является контекстно-свободным. Тогда язык $L - \{\varepsilon\}$ порождается некоторой контекстно-свободной грамматикой без ε-правил.*

Доказательство. Пусть дана контекстно-свободная грамматика $G = \langle N, \Sigma, P, S\rangle$, порождающая язык L. Проведём серию преобразований множества P.

Если для каких-то $A \in N$, $B \in N$, $\alpha \in (N \cup \Sigma)^*$ и $\beta \in (N \cup \Sigma)^*$ множество P содержит правила $B \to \alpha A \beta$ и $A \to \varepsilon$, но не содержит правила $B \to \alpha\beta$, то добавим это правило в P. Повторяем эту процедуру, пока возможно.

Теперь исключим из множества P все правила вида $A \to \varepsilon$. Полученная грамматика порождает язык $L - \{\varepsilon\}$. □

Пример 8.5. Рассмотрим язык L, порождаемый грамматикой $S \to \varepsilon$, $S \to aSbS$. Язык $L - \{\varepsilon\}$ порождается грамматикой $S \to aSbS$, $S \to abS$, $S \to aSb$, $S \to ab$.

8.3. Нормальная форма Хомского

[Сал, 6.2], [АхоУль, 2.4.2, 2.4.3], [ХопМотУль, 7.1], [ГорМол, с. 431–437], [Гла, 4.1], [СокКушБад, с. 32], [Лал, с. 330], [Sip, 2.1], [Рей, с. 66–68]

Определение 8.6. *Грамматика в нормальной форме Хомского* (*грамматика в бинарной нормальной форме, квадратичная грамматика*) (grammar in Chomsky normal form) — контекстно-свободная грамматика $\langle N, \Sigma, P, S\rangle$, в которой каждое правило имеет один из следующих трёх видов: $S \to \varepsilon$, $A \to a$, $A \to BC$, где $A \in N$, $B \in N - \{S\}$, $C \in N - \{S\}$, $a \in \Sigma$.

Пример 8.7. Грамматика $S \to RR$, $S \to AB$, $R \to RR$, $R \to AB$, $A \to a$, $B \to RB$, $B \to b$ — грамматика в нормальной форме Хомского.

Теорема 8.8. *Каждая контекстно-свободная грамматика эквивалентна некоторой грамматике в нормальной форме Хомского.*

Доказательство. Пусть дана контекстно-свободная грамматика $G = \langle N, \Sigma, P, S\rangle$. Проведём ряд преобразований этой грамматики так, что порождаемый ею язык остаётся неизменным.

Если правая часть какого-нибудь правила содержит символ S, то заменим грамматику $\langle N, \Sigma, P, S\rangle$ на грамматику $\langle N \cup \{S_0\}, \Sigma, P \cup \{S_0 \to S\}, S_0\rangle$, где S_0 — новый символ, не принадлежащий множеству $N \cup \Sigma$.

Заменим во всех правилах каждый терминальный символ a на новый нетерминальный символ T_a и добавим к множеству P правила $T_a \to a$ для всех $a \in \Sigma$.

Устраним правила вида $A \to \alpha$, где $|\alpha| > 2$, заменив каждое из них на ряд более коротких правил (при этом добавляются новые нетерминальные символы).

Теперь устраним все правила вида $A \to \varepsilon$, где A не является начальным символом. Это можно сделать, как в доказательстве теоремы 8.4.

Если для каких-то $A \in N$, $B \in N$ и $\alpha \in (N \cup \Sigma)^*$ множество P содержит правила $A \to B$ и $B \to \alpha$, но не содержит правила

$A \to \alpha$, то добавим это правило в P. Повторяем эту процедуру, пока возможно. После этого исключим из множества P все правила вида $A \to B$. □

Пример 8.9. Грамматика $S \to \varepsilon$, $S \to aUbU$, $U \to S$, $U \to ba$ эквивалентна следующей грамматике в нормальной форме Хомского: $S_0 \to \varepsilon$, $S_0 \to AD$, $D \to UC$, $D \to BU$, $D \to b$, $C \to BU$, $C \to b$, $U \to BA$, $U \to AD$, $A \to a$, $B \to b$.

Теорема 8.10. *Если контекстно-свободный язык не содержит пустого слова, то он порождается некоторой грамматикой, в которой каждое правило имеет один из следующих двух видов: $A \to a$, $A \to BC$, где $A \in N$, $B \in N - \{S\}$, $C \in N - \{S\}$, $a \in \Sigma$.*

8.4*. Нормальная форма Грейбах

[Сал, 6.2], [АхоУль, 2.4.4, 2.4.5], [ГорМол, с. 437–440], [Гла, 6.2], [Бра, с. 121–124], [Рей, с. 71–76], [СокКушБад, с. 32]

Определение 8.11. *Грамматика в нормальной форме Грейбах* (grammar in Greibach normal form) — контекстно-свободная грамматика $\langle N, \Sigma, P, S \rangle$, в которой каждое правило имеет один из следующих четырёх видов: $A \to \varepsilon$, $A \to a$, $A \to aB$, $A \to aBC$, где $A \in N$, $B \in N$, $C \in N$, $a \in \Sigma$.

Пример 8.12. Грамматика $S \to aST$, $S \to aT$, $T \to bS$, $T \to b$ — грамматика в нормальной форме Грейбах.

Замечание 8.13. Некоторые авторы разрешают в грамматиках в нормальной форме Грейбах использовать также правила вида $A \to a\beta$, где $A \in N$, $a \in \Sigma$, $\beta \in N^*$ (в определении 8.11 такие правила разрешены, только если $|\beta| \leqslant 2$).

Теорема 8.14. *Каждая контекстно-свободная грамматика эквивалентна некоторой грамматике в нормальной форме Грейбах.*

Доказательство. Докажем теорему для контекстно-свободных языков, не содержащих пустого слова. В силу теоремы 8.10 исходный язык порождается некоторой грамматикой $G = \langle N, \Sigma, P, S \rangle$, в которой каждое правило имеет вид $A \to a$ или $A \to BC$, где $A \in N$, $B \in N - \{S\}$, $C \in N - \{S\}$, $a \in \Sigma$.

Введём $|N|^2$ новых вспомогательных символов, соответствующих упорядоченным парам из множества $N \times N$. Новый символ, соответствующий паре $\langle A, B\rangle$, будем обозначать $(A\backslash B)$. Построим грамматику "почти в нормальной форме Грейбах" $\overline{G} = \langle \overline{N}, \Sigma, \overline{P}, S\rangle$, положив $\overline{N} = N \cup \{(A\backslash B) \mid A \in N,\ B \in N\}$ и

$$\begin{aligned}\overline{P} = {} & \{(A\backslash A) \to \varepsilon \mid A \in N\} \cup \\ & \cup \{(C\backslash E) \to A(A\backslash D)(B\backslash E) \mid (B \to CD) \in P,\ A \in N,\ E \in N\} \cup \\ & \cup \{A \to a \mid (A \to a) \in P\} \cup \\ & \cup \{S \to a(A\backslash S) \mid (A \to a) \in P\}.\end{aligned}$$

Если в этой грамматике заменить

$$\begin{aligned}\{(C\backslash E) \to A(A\backslash D)(B\backslash E) \mid (B \to CD) \in P,\ A \in N,\ E \in N\} \cup \\ \cup \{A \to a \mid (A \to a) \in P\}\end{aligned}$$

на

$$\{(C\backslash E) \to a(A\backslash D)(B\backslash E) \mid (B \to CD) \in P,\ (A \to a) \in P,\ E \in N\},$$

получим эквивалентную ей грамматику в нормальной форме Грейбах. Осталось лишь доказать, что $L(G) = L(\overline{G})$.

Сначала проверим индукцией по длине слова $\gamma \in N^*$, что если $F \overset{*}{\underset{G}{\Rightarrow}} C\gamma$, то $(C\backslash E) \overset{*}{\underset{\overline{G}}{\Rightarrow}} \gamma(F\backslash E)$ для любого $E \in N$. Чтобы провести шаг индукции, допустим, что $(B \to CD) \in P$, $F \overset{*}{\underset{G}{\Rightarrow}} B\beta \underset{G}{\Rightarrow} CD\beta \overset{*}{\underset{G}{\Rightarrow}} CA\alpha\beta$, где $D \overset{*}{\underset{G}{\Rightarrow}} A\alpha$ и $\gamma = A\alpha\beta$. По предположению индукции $(B\backslash E) \overset{*}{\underset{\overline{G}}{\Rightarrow}} \beta(F\backslash E)$ и $(A\backslash D) \overset{*}{\underset{\overline{G}}{\Rightarrow}} \alpha(D\backslash D)$. Подключая эти выводы к правилу $(C\backslash E) \underset{\overline{G}}{\Rightarrow} A(A\backslash D)(B\backslash E)$ и используя $(D\backslash D) \underset{\overline{G}}{\Rightarrow} \varepsilon$, получаем искомый вывод $(C\backslash E) \overset{*}{\underset{\overline{G}}{\Rightarrow}} A\alpha\beta(F\backslash E)$.

Докажем теперь, что для любого $\gamma \in N^*$ равносильны утверждения $F \overset{*}{\underset{G}{\Rightarrow}} C\gamma$ и $(C\backslash F) \overset{*}{\underset{\overline{G}}{\Rightarrow}} \gamma$. В одну сторону это следует из только что доказанного. Доказательство того, что если $(C\backslash F) \overset{*}{\underset{\overline{G}}{\Rightarrow}} \gamma$, то $F \overset{*}{\underset{G}{\Rightarrow}} C\gamma$, проведём индукцией по длине слова $\gamma \in N^*$. Чтобы провести шаг индукции, допустим, что $(B \to CD) \in P$, $(C\backslash F) \underset{\overline{G}}{\Rightarrow} A(A\backslash D)(B\backslash F)$, $(A\backslash D) \overset{*}{\underset{\overline{G}}{\Rightarrow}} \alpha$, $(B\backslash F) \overset{*}{\underset{\overline{G}}{\Rightarrow}} \beta$ и $\gamma = A\alpha\beta$. По предположению индукции $D \overset{*}{\underset{G}{\Rightarrow}} A\alpha$ и $F \overset{*}{\underset{G}{\Rightarrow}} B\beta$. Получаем искомый вывод $F \overset{*}{\underset{G}{\Rightarrow}} B\beta \underset{G}{\Rightarrow} CD\beta \overset{*}{\underset{G}{\Rightarrow}} CA\alpha\beta$.

Теперь убедимся, что $L(G) = L(\overline{G})$. Рассмотрим произвольное слово $a_0a_1\ldots a_m$, где $m \geqslant 0$ и $a_i \in \Sigma$ для всех $i \leqslant m$. Пусть $S \overset{*}{\underset{G}{\Rightarrow}} A_0A_1\ldots A_m \overset{*}{\underset{G}{\Rightarrow}} a_0a_1\ldots a_m$, где $A_i \in N$ для всех $i \leqslant m$. Тогда $S \underset{\overline{G}}{\Rightarrow} a_0(A_0\backslash S) \overset{*}{\underset{\overline{G}}{\Rightarrow}} a_0A_1\ldots A_m \overset{*}{\underset{\overline{G}}{\Rightarrow}} a_0a_1\ldots a_m$. Обратно, пусть $S \underset{\overline{G}}{\Rightarrow} a_0(A_0\backslash S) \overset{*}{\underset{\overline{G}}{\Rightarrow}} a_0A_1\ldots A_m \overset{*}{\underset{\overline{G}}{\Rightarrow}} a_0a_1\ldots a_m$, где $A_i \in N$ для всех $i \leqslant m$. Тогда $S \overset{*}{\underset{G}{\Rightarrow}} A_0A_1\ldots A_m \overset{*}{\underset{G}{\Rightarrow}} a_0a_1\ldots a_m$. □

Пример 8.15. Грамматика $S \to RT$, $T \to b$, $T \to UR$, $U \to VT$, $V \to RT$, $R \to a$ эквивалентна следующей грамматике в нормальной форме Грейбах: $S \to aC$, $C \to aD$, $C \to b$, $D \to aDE$, $D \to bE$, $E \to aDF$, $E \to bF$, $F \to a$. Здесь C, D, E и F соответствуют символам $(A\backslash S)$, $(A\backslash T)$, $(V\backslash T)$ и $(U\backslash T)$ из доказательства теоремы 8.14 (удалён 21 бесполезный символ).

Теорема 8.16. *Пусть язык L контекстно-свободный. Тогда язык $L - \{\varepsilon\}$ порождается некоторой грамматикой в нормальной форме Грейбах без ε-правил.*

Пример 8.17. Грамматика $S \to aR$, $R \to bRT$, $R \to \varepsilon$, $T \to cSR$, $T \to \varepsilon$ эквивалентна следующей грамматике в нормальной форме Грейбах без ε-правил: $S \to aR$, $S \to a$, $R \to bRT$, $R \to bT$, $R \to bR$, $R \to b$, $T \to cSR$, $T \to cS$.

9. Основные свойства контекстно-свободных языков

9.1. Лемма о разрастании для контекстно-свободных языков

[Sip, 2.3], [Гин, 3.1], [ХопМотУль, 7.2], [Лал, с. 299–300], [АхоУль, 2.6.1], [Гла, 4.3, 4.1], [ГорМол, с. 425–426], [СокКушБад, с. 33–34], [LewPap2, 3.5], [ГроЛан, 7.4.2], [Рей, с. 68–71], [КукБей, с. 279–281]

Лемма 9.1 (pumping lemma, лемма о разрастании, лемма о накачке, лемма-насос). *Пусть L — контекстно-свободный язык над алфавитом Σ. Тогда найдётся такое натуральное число p, что для любого слова $w \in L$ длины не меньше p найдутся слова $u, v, x, y, z \in \Sigma^*$, для которых верно $uvxyz = w$,*

$vy \neq \varepsilon$ (то есть $v \neq \varepsilon$ или $y \neq \varepsilon$), $|vxy| \leqslant p$ и $uv^ixy^iz \in L$ для всех $i \in \mathbb{N}$.

Доказательство. Пусть язык L порождается грамматикой в нормальной форме Хомского $G = \langle N, \Sigma, P, S\rangle$. Индукцией по k легко доказать, что для любого дерева вывода в грамматике G длина кроны дерева не превышает 2^{k-2}, где k — количество вершин в самом длинном пути, начинающемся в корне дерева и заканчивающемся в некоторой вершине, помеченной символом из Σ.

Положим $p = 2^{|N|}$. Пусть $w \in L$ и $|w| \geqslant p$. Зафиксируем некоторое дерево вывода с кроной w в грамматике G. Рассмотрим самый длинный путь в этом дереве. Этот путь содержит не менее $|N|+2$ вершин. Среди них найдутся две вершины с одинаковыми метками, причём их можно выбрать среди последних $|N|+2$ вершин рассматриваемого пути. Выберем слова u, v, x, y и z так, что $uvxyz = w$, поддерево с корнем в одной из найденных вершин имеет крону x и поддерево с корнем в другой найденной вершине имеет крону vxy.

Из того что G — грамматика в нормальной форме Хомского, заключаем, что $vxy \neq x$. Неравенство $|vxy| \leqslant 2^{|N|}$ следует из того, что самый длинный путь в соответствующем слову vxy поддереве содержит не более $|N|+2$ вершин. Для каждого $i \in \mathbb{N}$ можно построить дерево вывода с кроной uv^ixy^iz, комбинируя части исходного дерева вывода. □

Пример 9.2. Рассмотрим язык $L = \{a^nb^nc^n \mid n \geqslant 0\}$ над алфавитом $\{a, b, c\}$. Утверждение леммы 9.1 не выполняется ни для какого натурального числа p. Действительно, если $uvxyz = a^pb^pc^p$, $|vy| > 0$ и $|vxy| \leqslant p$, то $|vy|_a = 0$ или $|vy|_c = 0$. Следовательно, $|uvvxyyz|_a = p$ или $|uvvxyyz|_c = p$. Так как $|uvvxyyz| > 3p$, то $uvvxyyz \notin L$. Из этого можно заключить, что язык L не является контекстно-свободным.

Теорема 9.3. *Каждый контекстно-свободный язык над однобуквенным алфавитом является автоматным.*

Доказательство. Пусть дан контекстно-свободный язык L над алфавитом $\{a\}$. Согласно лемме 9.1 найдётся такое натуральное число p, что множество $\{k \in \mathbb{N} \mid a^k \in L\}$ является объединением некоторого семейства арифметических прогрессий, причём у каждой прогрессии первый член и шаг не больше числа p. Так как существует лишь конечное число прогрессий натуральных чисел с таким ограничением, то рассматриваемое семейство конечно. Следовательно, язык L является автоматным (используем пример 2.14). □

9.2. Свойства замкнутости класса линейных языков

[АхоУль, 2.6.4]

Пример 9.4. Пусть $\Sigma = \{a, b, c\}$. Язык $\{ucu^{\mathrm{R}} \mid u \in \{a, b\}^*\}$ является линейным, так как он порождается грамматикой $S \to aSa$, $S \to bSb$, $S \to c$.

Пример 9.5. Рассмотрим алфавит $\Sigma = \{a, b, c\}$. Язык $\{a^m b^n c^k \mid 0 \leqslant m < n,\ k \geqslant 0\}$ является линейным, так как он порождается грамматикой $S \to Sc$, $S \to T$, $T \to aTb$, $T \to Tb$, $T \to b$.

Теорема 9.6. *Если L_1 и L_2 — линейные языки над алфавитом Σ, то $L_1 \cup L_2$ тоже линейный язык.*

Доказательство. Пусть язык L_1 порождается грамматикой $\langle N_1, \Sigma, P_1, S_1\rangle$ и L_2 порождается грамматикой $\langle N_2, \Sigma, P_2, S_2\rangle$, где $N_1 \cap N_2 = \varnothing$. Тогда $L_1 \cup L_2$ порождается грамматикой $\langle N_1 \cup N_2 \cup \{T\}, \Sigma, P_1 \cup P_2 \cup \{T \to S_1,\ T \to S_2\}, T\rangle$, где $T \notin N_1 \cup N_2 \cup \Sigma$. □

Пример 9.7. Рассмотрим алфавит $\Sigma = \{a, b, c\}$. Язык $L = \Sigma^* - \{a^n b^n c^n \mid n \geqslant 0\}$ является линейным, поскольку $L = L_1 \cup L_2 \cup (\Sigma^* - L_3)$, где языки $L_1 = \{a^m b^n c^k \mid m \neq n,\ k \geqslant 0\}$ и $L_2 = \{a^m b^n c^k \mid n \neq k,\ m \geqslant 0\}$ являются линейными, а язык $L_3 = \{a^m b^n c^k \mid m \geqslant 0,\ n \geqslant 0,\ k \geqslant 0\}$ является автоматным, и можно применить теоремы 9.6, 3.8, 2.31 и лемму 1.68.

Упражнение 9.8. Пусть $\Sigma = \{a, b, c\}$. Является ли линейным язык $\Sigma^* - \{ucu \mid u \in \{a, b\}^*\}$?
Ответ. Да. Идею доказательства можно найти в [Гин, с. 106].

Упражнение 9.9. Пусть $\Sigma = \{a, b, c\}$. Является ли линейным язык $\Sigma^* - \{ucu^{\mathrm{R}} \mid u \in \{a, b\}^*\}$?
Ответ. Да.

9.3. Свойства замкнутости класса контекстно-свободных языков

[ГроЛан, 7.3.1–7.3.3], [Лал, с. 300–301], [ХопМотУль, 7.3.1–7.3.3], [АхоУль, 2.6.2], [Гин, 1.7], [Гла, 4.3], [ГорМол, с. 423–424], [LewPap2, 3.5]

Теорема 9.10. *Если L — контекстно-свободный язык, то L^* тоже контекстно-свободный язык.*

Доказательство. Пусть язык L порождается грамматикой $\langle N, \Sigma, P, S\rangle$. Тогда язык L^* порождается грамматикой $\langle N \cup \{T\}, \Sigma, P \cup \{T \to ST,\ T \to \varepsilon\}, T\rangle$, где $T \notin N \cup \Sigma$. □

Теорема 9.11. *Если L_1 и L_2 — контекстно-свободные языки над алфавитом Σ, то $L_1 \cdot L_2$ тоже контекстно-свободный язык.*

Доказательство. Пусть язык L_1 порождается грамматикой $\langle N_1, \Sigma, P_1, S_1\rangle$ и L_2 порождается грамматикой $\langle N_2, \Sigma, P_2, S_2\rangle$, где $N_1 \cap N_2 = \varnothing$. Тогда $L_1 \cdot L_2$ порождается грамматикой $\langle N_1 \cup N_2 \cup \{T\}, \Sigma, P_1 \cup P_2 \cup \{T \to S_1S_2\}, T\rangle$, где $T \notin N_1 \cup N_2 \cup \Sigma$. □

Теорема 9.12. *Если L_1 и L_2 — контекстно-свободные языки над алфавитом Σ, то $L_1 \cup L_2$ тоже контекстно-свободный язык.*

Доказательство. Пусть язык L_1 порождается грамматикой $\langle N_1, \Sigma, P_1, S_1\rangle$ и L_2 порождается грамматикой $\langle N_2, \Sigma, P_2, S_2\rangle$, где $N_1 \cap N_2 = \varnothing$. Тогда $L_1 \cup L_2$ порождается грамматикой $\langle N_1 \cup N_2 \cup \{T\}, \Sigma, P_1 \cup P_2 \cup \{T \to S_1,\ T \to S_2\}, T\rangle$, где $T \notin N_1 \cup N_2 \cup \Sigma$. □

Теорема 9.13. *Если L — контекстно-свободный язык, то L^{R} тоже контекстно-свободный язык.*

9.4. Пересечение и дополнение контекстно-свободных языков

[Гин, 3.2], [АхоУль, 2.6.2], [ХопМотУль, 7.3.4], [Гла, 4.3], [Лал, с. 300–301], [LewPap2, 3.5], [ГроЛан, 7.3.4, 7.3.5], [ГорМол, с. 424], [Рей, с. 87–88]

Теорема 9.14. *Неверно, что для любых контекстно-свободных языков L_1 и L_2 язык $L_1 \cap L_2$ тоже контекстно-свободный.*

Доказательство. Положим $L_1 = \{a^m b^m c^n \mid m \geqslant 0,\ n \geqslant 0\}$ и $L_2 = \{a^m b^n c^n \mid m \geqslant 0,\ n \geqslant 0\}$. В примере 9.2 было доказано, что язык $L_1 \cap L_2$ не является контекстно-свободным. □

Теорема 9.15. *Неверно, что для любого контекстно-свободного языка $L \subseteq \Sigma^*$ язык $\Sigma^* - L$ тоже контекстно-свободный.*

Доказательство. Положим $L = \Sigma^* - \{a^n b^n c^n \mid n \geqslant 0\}$, где $\Sigma = \{a, b, c\}$. В примере 9.7 было доказано, что язык L является линейным (и следовательно, контекстно-свободным). □

9.5. Пересечение контекстно-свободного языка с автоматным языком

[Гин, 3.2], [АхоУль, 2.6.2], [ХопМотУль, 7.3.4], [Гла, 5.2], [Лал, с. 301], [LewPap2, 3.5], [ГроЛан, 10.6.3], [ТраБар, 1.12], [Рей, с. 87–88]

Теорема 9.16. *Если L_1 — контекстно-свободный язык и L_2 — автоматный язык, то язык $L_1 \cap L_2$ является контекстно-свободным.*

Доказательство. Пусть $\langle N, \Sigma, P, S\rangle$ — контекстно-свободная грамматика, порождающая язык L_1. Без ограничения общности можно считать, что множество P содержит только правила вида $A \to a$ и $A \to \alpha$, где $A \in N$, $a \in \Sigma$ и $\alpha \in N^*$ (в силу теоремы 8.8). Пусть $\langle Q, \Sigma, \Delta, I, F\rangle$ — конечный автомат, распознающий язык L_2. Без ограничения общности можно считать, что для каждого перехода $\langle p, x, q\rangle \in \Delta$ выполняется равенство $|x| = 1$ (см. лемму 2.30).

Построим контекстно-свободную грамматику $\langle \overline{N}, \overline{\Sigma}, \overline{P}, \overline{S}\rangle$, порождающую язык $L_1 \cap L_2$. Положим

$$\begin{aligned}
&\overline{N} = \{\overline{S}\} \cup (Q \times N \times Q),\\
&\overline{P} = \{\overline{S} \to \langle p, S, q\rangle \mid p \in I,\ q \in F\} \cup\\
&\quad \cup \{\langle p, A, q\rangle \to a \mid \langle p, a, q\rangle \in \Delta,\ (A \to a) \in P\} \cup\\
&\quad \cup \{\langle p_0, A, p_n\rangle \to \langle p_0, B_1, p_1\rangle \ldots \langle p_{n-1}, B_n, p_n\rangle \mid\\
&\qquad (A \to B_1 \ldots B_n) \in P,\ p_0 \in Q, \ldots, p_n \in Q\},
\end{aligned}$$

где $\overline{S}$ — новый символ (не принадлежащий множеству $Q \times N \times Q$). □

Пример 9.17. Пусть $\Sigma = \{a, b, c\}$. Рассмотрим контекстно-свободный язык L_1, порождаемый грамматикой $S \to a$, $S \to bS$, $S \to cSS$, и автоматный язык L_2, заданный конечным автоматом $M = \langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2\}$, $\Delta = \{\langle 1, a, 2\rangle, \langle 1, c, 2\rangle, \langle 2, a, 1\rangle, \langle 2, b, 1\rangle\}$, $I = \{1\}$ и $F = \{2\}$. Тогда язык $L_1 \cap L_2$ порождается контекстно-свободной грамматикой $\overline{S} \to S_{12}$, $S_{12} \to a$, $S_{21} \to a$, $S_{21} \to bS_{11}$, $S_{22} \to bS_{12}$, $S_{11} \to cS_{21}S_{11}$, $S_{11} \to cS_{22}S_{21}$, $S_{12} \to cS_{21}S_{12}$, $S_{12} \to cS_{22}S_{22}$. Здесь S_{11}, S_{12}, S_{21} и S_{22} соответствуют символам $\langle 1, S, 1\rangle$, $\langle 1, S, 2\rangle$, $\langle 2, S, 1\rangle$ и $\langle 2, S, 2\rangle$ из доказательства теоремы 9.16.

Замечание 9.18*. Если L_1 — линейный язык и L_2 — автоматный язык, то язык $L_1 \cap L_2$ является линейным.

Пример 9.19*. Пусть $\Sigma = \{a, b\}$. Рассмотрим линейный язык L_1, порождаемый грамматикой $S \to aaSaa$, $S \to bSb$, $S \to \varepsilon$, и автоматный язык L_2, заданный конечным автоматом $M = \langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2, 3\}$, $\Delta = \{\langle 1, a, 1\rangle, \langle 1, b, 1\rangle, \langle 1, a, 2\rangle, \langle 2, b, 3\rangle, \langle 3, a, 3\rangle, \langle 3, b, 3\rangle\}$, $I = \{1\}$ и $F = \{3\}$. Тогда язык $L_1 \cap L_2$ порождается контекстно-свободной грамматикой $\overline{S} \to S_{13}$, $S_{11} \to aaS_{11}aa$, $S_{12} \to aaS_{11}aa$, $S_{13} \to aaS_{13}aa$, $S_{13} \to aaS_{23}aa$, $S_{33} \to aaS_{33}aa$, $S_{11} \to bS_{11}b$, $S_{13} \to bS_{13}b$, $S_{13} \to bS_{12}b$, $S_{23} \to bS_{33}b$, $S_{33} \to bS_{33}b$, $S_{11} \to \varepsilon$, $S_{33} \to \varepsilon$. Эту грамматику можно упростить, заменив S_{11} и S_{33} на один символ.

9.6*. Теорема Парика

[Гин, 5.1, 5.2], [Гла, 4.3], [LewPap1, 3.5.2], [АхоУль, с. 239]

Замечание 9.20. В этом разделе предполагается, что зафиксирован некоторый линейный порядок на алфавите Σ. Пусть $\Sigma = \{a_1, \ldots, a_n\}$.

Определение 9.21. Через Ψ_Σ будем обозначать функцию из Σ^* в $\mathbb{N}^n$, определённую следующим образом: $\Psi_\Sigma(w) \rightleftharpoons \langle |w|_{a_1}, \ldots, |w|_{a_n}\rangle$. Аналогично, каждому языку $L \subseteq \Sigma^*$ ставится в соответствие множество $\Psi_\Sigma(L) \subseteq \mathbb{N}^n$, определённое так: $\Psi_\Sigma(L) \rightleftharpoons \{\Psi_\Sigma(w) \mid w \in L\}$.

Пример 9.22. Пусть $\Sigma = \{a_1, a_2\}$ и $L = \{a_1, a_1a_2a_2, a_2a_2a_1\}$. Тогда $\Psi_\Sigma(L) = \{\langle 1, 0\rangle, \langle 1, 2\rangle\}$.

Определение 9.23. Пусть $B \subseteq \mathbb{N}^n$ и $P \subseteq \mathbb{N}^n$. Тогда через $\mathrm{L}(B, P)$ обозначается множество

$$\{b + p_1 + \ldots + p_k \mid b \in B,\ k \geqslant 0,\ p_1, \ldots, p_k \in P\}.$$

Множество B называется *системой предпериодов* множества $\mathrm{L}(B, P)$. Множество P называется *системой периодов* множества $\mathrm{L}(B, P)$.

Определение 9.24. Множество $A \subseteq \mathbb{N}^n$ называется *линейным* (linear), если $A = \mathrm{L}(B, P)$ для некоторых конечных множеств B и P.

Определение 9.25. Множество $A \subseteq \mathbb{N}^n$ называется *полулинейным* (semilinear), если оно является объединением конечного числа линейных множеств.

Теорема 9.26 (Теорема Пари́ка). *Если язык $L \subseteq \Sigma^*$ является контекстно-свободным, то множество $\Psi_\Sigma(L)$ является полулинейным.*

Доказательство можно найти в [Гин, с. 207–211].

Пример 9.27. Пусть $\Sigma = \{a, b\}$. Рассмотрим язык $L = \{a^m b^n \mid m > n \text{ или } m \text{ простое}\}$. Можно проверить, что множество $\Psi_\Sigma(L)$ не является полулинейным. Следовательно, язык L не является контекстно-свободным.

Теорема 9.28. *Если множество $A \subseteq \mathbb{N}^n$ является полулинейным, то существует такой автоматный язык L, что $A = \Psi_\Sigma(L)$.*

Доказательство. Докажем это для произвольного линейного множества $A = \mathrm{L}(B, P)$ (на полулинейные множества утверждение распространяется по теореме 3.1). Рассмотрим конечный автомат $M = \langle Q, \Sigma, \Delta, I, F\rangle$, где $Q = \{1, 2\}$, $I = \{1\}$, $F = \{2\}$ и

$$\Delta = \{\langle 1, a_1^{k_1} \ldots a_n^{k_n}, 2\rangle \mid \langle k_1, \ldots, k_n\rangle \in B\} \cup \\ \cup \{\langle 2, a_1^{k_1} \ldots a_n^{k_n}, 2\rangle \mid \langle k_1, \ldots, k_n\rangle \in P\}.$$

Очевидно, $\Psi_\Sigma(L(M)) = \mathrm{L}(B, P)$. □

Замечание 9.29. Теорема 9.3 является следствием теорем 9.26 и 9.28.

10. Автоматы с магазинной памятью

10.1. Определение автомата с магазинной памятью

[Гин, 2.4], [АхоУль, 2.5.1, 2.5.2], [ХопМотУль, 6.1, 6.2], [Гла, 4.5], [ГлаМел, с. 136–149], [ГорМол, с. 418–420], [СокКушБад, с. 36–38], [ЛПИИ, 5.2.5, 5.2.6], [Бра, с. 109–116], [ГроЛан, 9.1, 9.2], [LewPap2, 3.3], [Sip, 2.2], [Рей, с. 79–83]

Определение 10.1. *Автомат с магазинной памятью* (*МП-автомат, магазинный автомат, стековый автомат*) (pushdown automaton) — это шестёрка $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$, где Q, Σ, Γ и Δ — конечные множества, $I \subseteq Q$, $F \subseteq Q$ и $\Delta \subseteq (Q \times \Sigma^* \times \Gamma^*) \times (Q \times \Gamma^*)$. Здесь Q — *множество состояний*, Σ — *входной алфавит*, Γ — *алфавит магазинной памяти* (stack alphabet), Δ — *множество переходов* (transition relation), элементы I называются *начальными* состояниями, элементы F — *заключительными* или *допускающими* состояниями.

Пример 10.2. Пусть $Q = \{1, 2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{A, B\}$,

$$\Delta = \{\langle\langle 1, a, \varepsilon\rangle, \langle 1, A\rangle\rangle,\ \langle\langle 1, b, \varepsilon\rangle, \langle 1, B\rangle\rangle,\ \langle\langle 1, \varepsilon, \varepsilon\rangle, \langle 2, \varepsilon\rangle\rangle,\\ \langle\langle 2, a, A\rangle, \langle 2, \varepsilon\rangle\rangle,\ \langle\langle 2, b, B\rangle, \langle 2, \varepsilon\rangle\rangle\},$$

$I = \{1\}$, $F = \{2\}$. Тогда $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ — МП-автомат.

Определение 10.3. *Конфигурацией* МП-автомата называется любая тройка $\langle q, w, \alpha \rangle$, где $q \in Q$, $w \in \Sigma^*$, $\alpha \in \Gamma^*$.

Определение 10.4. Определим на множестве всех конфигураций МП-автомата $M = \langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ бинарное отношение $\underset{M}{\vdash}$ (*такт работы*) следующим образом. Если $\langle\langle p, x, \beta\rangle, \langle q, \gamma\rangle\rangle \in \Delta$, $w \in \Sigma^*$ и $\alpha \in \Gamma^*$, то $\langle p, xw, \beta\alpha \rangle \underset{M}{\vdash} \langle q, w, \gamma\alpha \rangle$.

Замечание 10.5. Обычно из контекста ясно, о каком МП-автомате идёт речь. Тогда вместо $\underset{M}{\vdash}$ будем писать $\vdash$.

Пример 10.6. Для МП-автомата $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$ из примера 10.2 выполняется $\langle 1, abba, \varepsilon \rangle \vdash \langle 1, bba, A \rangle$ и $\langle 1, abba, \varepsilon \rangle \vdash \langle 2, abba, \varepsilon \rangle$.

Определение 10.7. Бинарное отношение $\overset{*}{\vdash}$ определяется как рефлексивное, транзитивное замыкание отношения $\vdash$.

Пример 10.8. Для МП-автомата $\langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$ из примера 10.2 выполняется $\langle 1, abba, \varepsilon\rangle \overset{*}{\vdash} \langle 1, ba, BA\rangle$ и $\langle 1, abba, \varepsilon\rangle \overset{*}{\vdash} \langle 2, \varepsilon, \varepsilon\rangle$.

Лемма 10.9. *Если* $\langle p, x, \alpha_1\rangle \overset{*}{\vdash} \langle q, \varepsilon, \alpha_2\rangle$ *и* $\langle q, y, \alpha_2\alpha_3\rangle \overset{*}{\vdash} \langle r, \varepsilon, \alpha_4\rangle$, *то* $\langle p, xy, \alpha_1\alpha_3\rangle \overset{*}{\vdash} \langle r, \varepsilon, \alpha_4\rangle$.

Доказательство. Индукция по количеству тактов в вычислении, ведущем из конфигурации $\langle p, x, \alpha_1\rangle$ в конфигурацию $\langle q, \varepsilon, \alpha_2\rangle$. □

Определение 10.10. Слово $w \in \Sigma^*$ *допускается* МП-автоматом $M = \langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$, если найдутся такие состояния $s \in I$ и $q \in F$, что $\langle s, w, \varepsilon\rangle \overset{*}{\vdash} \langle q, \varepsilon, \varepsilon\rangle$.

Определение 10.11. Язык, *распознаваемый* МП-автоматом, — множество всех слов, допускаемых этим МП-автоматом. Язык, распознаваемый МП-автоматом M, обозначается $L(M)$.

Замечание 10.12. Обычно в учебниках используется более узкое определение МП-автомата, но это не меняет класса языков, распознаваемых МП-автоматами.

Пример 10.13. Пусть M — МП-автомат из примера 10.2. Тогда $L(M) = \{uu^{\mathrm{R}} \mid u \in \{a,b\}^*\}$.

Пример 10.14. Пусть $\Sigma = \{a, b, c\}$. Рассмотрим МП-автомат $M = \langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$, где $Q = \{1, 2, 3, 4, 5\}$, $\Gamma = \{A, B\}$, $I = \{1\}$, $F = \{4, 5\}$ и

$$
\begin{aligned}
\Delta = \{&\langle\langle 1, a, \varepsilon\rangle, \langle 1, A\rangle\rangle,\ \langle\langle 1, b, \varepsilon\rangle, \langle 1, B\rangle\rangle,\\
&\langle\langle 1, c, \varepsilon\rangle, \langle 2, \varepsilon\rangle\rangle,\\
&\langle\langle 2, a, A\rangle, \langle 2, \varepsilon\rangle\rangle,\ \langle\langle 2, b, B\rangle, \langle 2, \varepsilon\rangle\rangle,\\
&\langle\langle 2, a, B\rangle, \langle 3, \varepsilon\rangle\rangle,\ \langle\langle 2, b, A\rangle, \langle 3, \varepsilon\rangle\rangle,\\
&\langle\langle 2, \varepsilon, A\rangle, \langle 4, \varepsilon\rangle\rangle,\ \langle\langle 2, \varepsilon, B\rangle, \langle 4, \varepsilon\rangle\rangle,\\
&\langle\langle 2, b, \varepsilon\rangle, \langle 5, \varepsilon\rangle\rangle,\ \langle\langle 2, a, \varepsilon\rangle, \langle 5, \varepsilon\rangle\rangle,\\
&\langle\langle 3, a, \varepsilon\rangle, \langle 3, \varepsilon\rangle\rangle,\ \langle\langle 3, b, \varepsilon\rangle, \langle 3, \varepsilon\rangle\rangle,\\
&\langle\langle 3, \varepsilon, \varepsilon\rangle, \langle 4, \varepsilon\rangle\rangle,\\
&\langle\langle 4, \varepsilon, A\rangle, \langle 4, \varepsilon\rangle\rangle,\ \langle\langle 4, \varepsilon, B\rangle, \langle 4, \varepsilon\rangle\rangle,\\
&\langle\langle 5, a, \varepsilon\rangle, \langle 5, \varepsilon\rangle\rangle,\ \langle\langle 5, b, \varepsilon\rangle, \langle 5, \varepsilon\rangle\rangle\}.
\end{aligned}
$$

Тогда $L(M) = \{ucv \mid u \in \{a,b\}^*,\ v \in \{a,b\}^*,\ u \neq v^{\mathrm{R}}\}$.

Определение 10.15. Два МП-автомата *эквивалентны*, если они распознают один и тот же язык.

Замечание 10.16. В данном пособии не рассматриваются *преобразователи с магазинной памятью* (pushdown transducer) — обобщение автоматов с магазинной памятью посредством добавления “выходного” потока (см. [Гин, 3.5] или [АхоУль, 3.1.4]).

Замечание 10.17. Некоторые авторы изменяют определение допускаемых слов следующим образом: слово w допускается МП-автоматом $\langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$, если найдутся такие состояния $s \in I$ и $q \in F$ и последовательность $\alpha \in \Gamma^*$, что $\langle s, w, \varepsilon\rangle \overset{*}{\vdash} \langle q, \varepsilon, \alpha\rangle$. Такое определение не изменяет класса языков, распознаваемых МП-автоматами.

Замечание 10.18. Некоторые авторы добавляют в определение МП-автомата седьмую компоненту — Z_0, называемую *маркером магазинной памяти* (start pushdown symbol), и изменяют определение допускаемых слов следующим образом: слово w допускается МП-автоматом $\langle Q, \Sigma, \Gamma, \Delta, I, F, Z_0\rangle$, если найдутся такие состояния $s \in I$ и $q \in F$, что $\langle s, w, Z_0\rangle \overset{*}{\vdash} \langle q, \varepsilon, \varepsilon\rangle$. Такое определение не изменяет класса языков, распознаваемых МП-автоматами.

Замечание 10.19. Класс языков, распознаваемых МП-автоматами, не изменится также, если использовать следующую естественную комбинацию двух предыдущих определений: слово w допускается МП-автоматом $\langle Q, \Sigma, \Gamma, \Delta, I, F, Z_0\rangle$, если найдутся такие состояния $s \in I$ и $q \in F$ и последовательность $\alpha \in \Gamma^*$, что $\langle s, w, Z_0\rangle \overset{*}{\vdash} \langle q, \varepsilon, \alpha\rangle$.

Замечание 10.20. Некоторые авторы добавляют в определение МП-автомата маркер магазинной памяти Z_0, отбрасывают множество F и изменяют определение допускаемых слов следующим образом: слово w допускается МП-автоматом $\langle Q, \Sigma, \Gamma, \Delta, I, Z_0\rangle$, если найдутся такие состояния $s \in I$ и $q \in Q$, что $\langle s, w, Z_0\rangle \overset{*}{\vdash} \langle q, \varepsilon, \varepsilon\rangle$. И это не изменяет класса языков, распознаваемых МП-автоматами.

10.2. Характеризация контекстно-свободных языков

[Sip, 2.2], [Гин, 2.5], [АхоУль, 2.5.3], [ХопМотУль, 6.3], [Гла, 4.5], [ГлаМел, с. 149–150], [ГорМол, с. 420–422], [Бра, с. 116–120], [ГроЛан, 9.3, 15.3], [LewPap2, 3.4], [Рей, с. 84–87]

Теорема 10.21. *Если язык L является контекстно-свободным, то существует МП-автомат, распознающий этот язык.*

Доказательство. Пусть язык L порождается грамматикой $\langle N, \Sigma, P, S\rangle$, в которой каждое правило имеет вид $A \to w\beta$, где $A \in N$, $w \in \Sigma^*$ и $\beta \in N^*$ (в силу теоремы 8.8 такая грамматика существует). Положим $\Gamma = N$, $Q = \{1, 2\}$, $I = \{1\}$, $F = \{2\}$ и

$$\Delta = \{\langle\langle 1, \varepsilon, \varepsilon\rangle, \langle 2, S\rangle\rangle\} \cup \{\langle\langle 2, w, A\rangle, \langle 2, \beta\rangle\rangle \mid (A \to w\beta) \in P\}.$$

Можно доказать, что $\langle 2, u, S\rangle \overset{*}{\vdash} \langle 2, \varepsilon, \alpha\rangle$ тогда и только тогда, когда существует левосторонний вывод $S \underset{\text{lm}}{\overset{*}{\Rightarrow}} u\alpha$ (здесь $u \in \Sigma^*$ и $\alpha \in N^*$). □

Пример 10.22. Пусть $\Sigma = \{a, b, c, d\}$. Контекстно-свободная грамматика $S \to SS$, $S \to a$, $S \to bcTS$, $T \to cSS$, $T \to dTT$ и МП-автомат $\langle\{1, 2\}, \Sigma, \{S, T\}, \Delta, \{1\}, \{2\}\rangle$, где

$$\begin{aligned}\Delta = \{&\langle\langle 1, \varepsilon, \varepsilon\rangle, \langle 2, S\rangle\rangle,\ \langle\langle 2, \varepsilon, S\rangle, \langle 2, SS\rangle\rangle,\ \langle\langle 2, a, S\rangle, \langle 2, \varepsilon\rangle\rangle,\\ &\langle\langle 2, bc, S\rangle, \langle 2, TS\rangle\rangle,\ \langle\langle 2, c, T\rangle, \langle 2, SS\rangle\rangle,\ \langle\langle 2, d, T\rangle, \langle 2, TT\rangle\rangle\},\end{aligned}$$

задают один и тот же язык.

Лемма 10.23. *Каждый МП-автомат эквивалентен некоторому МП-автомату $\langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$, где $|I| = 1$, $|F| = 1$ и каждый переход $\langle\langle p, x, \beta\rangle, \langle q, \gamma\rangle\rangle \in \Delta$ удовлетворяет требованиям $|x| \leqslant 1$ и $|\beta| + |\gamma| = 1$.*

Пример 10.24. Рассмотрим МП-автомат $M = \langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$, где $Q = I = F = \{1\}$, $\Sigma = \{a, b, c, d\}$, $\Gamma = \{A, B\}$,

$$\begin{aligned}\Delta = \{&\langle\langle 1, a, A\rangle, \langle 1, \varepsilon\rangle\rangle,\ \langle\langle 1, b, B\rangle, \langle 1, \varepsilon\rangle\rangle,\\ &\langle\langle 1, c, \varepsilon\rangle, \langle 1, A\rangle\rangle,\ \langle\langle 1, d, A\rangle, \langle 1, AB\rangle\rangle\}.\end{aligned}$$

Он эквивалентен МП-автомату $M' = \langle Q', \Sigma, \Gamma, \Delta', I, F\rangle$, где $Q' = \{1, 2, 3\}$ и

$$\Delta' = \{\langle\langle 1, a, A\rangle, \langle 1, \varepsilon\rangle\rangle,\ \langle\langle 1, b, B\rangle, \langle 1, \varepsilon\rangle\rangle,\ \langle\langle 1, c, \varepsilon\rangle, \langle 1, A\rangle\rangle,$$
$$\langle\langle 1, d, A\rangle, \langle 2, \varepsilon\rangle\rangle,\ \langle\langle 2, \varepsilon, \varepsilon\rangle, \langle 3, B\rangle\rangle,\ \langle\langle 3, \varepsilon, \varepsilon\rangle, \langle 1, A\rangle\rangle\}.$$

Пример 10.25. Рассмотрим МП-автомат $M = \langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$, где $Q = I = F = \{1\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{A\}$, $\Delta = \{\langle\langle 1, a, \varepsilon\rangle, \langle 1, A\rangle\rangle,\ \langle\langle 1, b, A\rangle, \langle 1, \varepsilon\rangle\rangle,\ \langle\langle 1, c, \varepsilon\rangle, \langle 1, \varepsilon\rangle\rangle\}$. Он эквивалентен МП-автомату $M' = \langle Q', \Sigma, \Gamma', \Delta', I, F\rangle$, где $Q' = \{1, 2\}$, $\Gamma' = \{A, T\}$ и

$$\Delta' = \{\langle\langle 1, a, \varepsilon\rangle, \langle 1, A\rangle\rangle,\ \langle\langle 1, b, A\rangle, \langle 1, \varepsilon\rangle\rangle,$$
$$\langle\langle 1, c, \varepsilon\rangle, \langle 2, T\rangle\rangle,\ \langle\langle 2, \varepsilon, T\rangle, \langle 1, \varepsilon\rangle\rangle\}.$$

Пример 10.26. Рассмотрим МП-автомат $M = \langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$, где $Q = I = F = \{1, 2\}$, $\Sigma = \{a, b\}$, $\Gamma = \{C\}$,

$$\Delta = \{\langle\langle 1, a, \varepsilon\rangle, \langle 1, C\rangle\rangle,\ \langle\langle 1, b, C\rangle, \langle 1, \varepsilon\rangle\rangle,$$
$$\langle\langle 2, b, \varepsilon\rangle, \langle 2, C\rangle\rangle,\ \langle\langle 2, a, C\rangle, \langle 2, \varepsilon\rangle\rangle\}.$$

Он эквивалентен МП-автомату $\langle\{0, 1, 2, 3\}, \Sigma, \Gamma, \Delta', \{0\}, \{3\}\rangle$, где

$$\Delta' = \Delta \cup \{\langle\langle 0, \varepsilon, \varepsilon\rangle, \langle 1, \varepsilon\rangle\rangle,\ \langle\langle 0, \varepsilon, \varepsilon\rangle, \langle 2, \varepsilon\rangle\rangle,$$
$$\langle\langle 1, \varepsilon, \varepsilon\rangle, \langle 3, \varepsilon\rangle\rangle,\ \langle\langle 2, \varepsilon, \varepsilon\rangle, \langle 3, \varepsilon\rangle\rangle\}.$$

Теорема 10.27. *Если язык L распознаётся некоторым МП-автоматом, то L является контекстно-свободным.*

Доказательство. Пусть язык L распознаётся МП-автоматом $\langle Q, \Sigma, \Gamma, \Delta, I, F\rangle$. Без ограничения общности можно считать, что $I = \{q_{\mathrm{s}}\}$, $F = \{q_{\mathrm{a}}\}$ и каждый переход $\langle\langle p, x, \beta\rangle, \langle q, \gamma\rangle\rangle \in \Delta$ удовлетворяет требованию $|\beta| + |\gamma| = 1$. Построим искомую контекстно-свободную грамматику $\langle N, \Sigma, P, S\rangle$, положив $N = \{A_{p,q} \mid p \in Q,\ q \in Q\}$, $S = A_{q_{\mathrm{s}}, q_{\mathrm{a}}}$ и

$$P = \{A_{p,p} \to \varepsilon \mid p \in Q\} \cup$$
$$\cup \{A_{p,t} \to xA_{q,r}yA_{s,t} \mid \langle\langle p, x, \varepsilon\rangle, \langle q, C\rangle\rangle \in \Delta,$$
$$\langle\langle r, y, C\rangle, \langle s, \varepsilon\rangle\rangle \in \Delta,\ C \in \Gamma,\ t \in Q\}.$$

Можно доказать, что $\langle p, x, \varepsilon\rangle \overset{*}{\vdash} \langle q, \varepsilon, \varepsilon\rangle$ тогда и только тогда, когда $A_{p,q} \overset{*}{\Rightarrow} x$ (здесь $x \in \Sigma^*$). $\square$

Пример 10.28. МП-автомат $M = \langle \{1,2\}, \Sigma, \Gamma, \Delta, \{1\}, \{2\} \rangle$, где $\Sigma = \{a, b\}$, $\Gamma = \{D, E\}$,

$$\Delta = \{ \langle \langle 1, ab, \varepsilon \rangle, \langle 1, E \rangle \rangle,\ \langle \langle 1, aa, E \rangle, \langle 1, \varepsilon \rangle \rangle,\ \langle \langle 1, b, \varepsilon \rangle, \langle 2, D \rangle \rangle,\\ \langle \langle 2, a, \varepsilon \rangle, \langle 1, D \rangle \rangle,\ \langle \langle 2, \varepsilon, D \rangle, \langle 2, \varepsilon \rangle \rangle,\ \langle \langle 2, b, E \rangle, \langle 2, \varepsilon \rangle \rangle \},$$

и контекстно-свободная грамматика $S \to abTaaS$, $S \to abSbU$, $S \to bUU$, $T \to abTaaT$, $T \to \varepsilon$, $U \to aSU$, $U \to \varepsilon$ задают один и тот же язык. Здесь S, T и U соответствуют символам $A_{1,2}$, $A_{1,1}$ и $A_{2,2}$ из доказательства теоремы 10.27.

10.3*. Автоматы с магазинной памятью с однобуквенными переходами

[АхоУль, с. 218], [Гла, 6.2], [Бра, с. 124–125]

Теорема 10.29. *Каждый МП-автомат эквивалентен некоторому МП-автомату $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$, где $|Q| = 2$ и каждый переход $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$ удовлетворяет требованиям $|x| = 1$, $|\beta| \leqslant 1$ и $|\gamma| \leqslant 2$.*

Доказательство. Пусть исходным МП-автоматом распознаётся контекстно-свободный язык $L \subseteq \Sigma^*$. Согласно теореме 8.16 язык $L - \{\varepsilon\}$ порождается некоторой контекстно-свободной грамматикой $\langle N, \Sigma, P, S \rangle$, в которой каждое правило имеет вид $A \to a\gamma$, где $A \in N$, $a \in \Sigma$, $\gamma \in N^*$ и $|\gamma| \leqslant 2$. Аналогично тому, как было сделано при доказательстве теоремы 10.21, положим $\Gamma = N$, $Q = \{1,2\}$, $I = \{1\}$, $\Delta = \{ \langle \langle 2, a, A \rangle, \langle 2, \gamma \rangle \rangle \mid (A \to a\gamma) \in P \} \cup \{ \langle \langle 1, a, \varepsilon \rangle, \langle 2, \gamma \rangle \rangle \mid (S \to a\gamma) \in P \}$,

$$F = \begin{cases} \{1,2\}, & \text{если } \varepsilon \in L, \\ \{2\}, & \text{если } \varepsilon \notin L. \end{cases}$$

□

Теорема 10.30. *Каждый МП-автомат эквивалентен некоторому МП-автомату $\langle Q, \Sigma, \Gamma, \Delta, I, F \rangle$, в котором каждый переход $\langle \langle p, x, \beta \rangle, \langle q, \gamma \rangle \rangle \in \Delta$ удовлетворяет требованиям $|x| = 1$, $|\beta| \leqslant 1$ и $|\gamma| \leqslant 1$.*

Доказательство. Пусть исходным МП-автоматом распознаётся контекстно-свободный язык $L \subseteq \Sigma^*$. Согласно теореме 8.16

язык $L - \{\varepsilon\}$ порождается некоторой контекстно-свободной грамматикой $G = \langle N, \Sigma, P, S\rangle$, в которой каждое правило имеет один из следующих трёх видов: $A \to a$, $A \to aB$, $A \to aBC$, где $A \in N$, $B \in N$, $C \in N$, $a \in \Sigma$. Легко добиться того, чтобы $B \neq S$ и $C \neq S$ в правилах грамматики G.

Положим $Q = N \cup \{\perp\}$, где $\perp \notin N$. Далее, положим $\Gamma = N$,

$$I = \{S\}, \qquad F = \begin{cases} \{S, \perp\}, & \text{если } \varepsilon \in L, \\ \{\perp\}, & \text{если } \varepsilon \notin L, \end{cases}$$

$$\begin{aligned} \Delta = {} & \{\langle\langle A, a, \varepsilon\rangle, \langle B, C\rangle\rangle \mid (A \to aBC) \in P\} \cup \\ & \cup \{\langle\langle A, a, \varepsilon\rangle, \langle B, \varepsilon\rangle\rangle \mid (A \to aB) \in P\} \cup \\ & \cup \{\langle\langle A, a, D\rangle, \langle D, \varepsilon\rangle\rangle \mid (A \to a) \in P,\ D \in N\} \cup \\ & \cup \{\langle\langle A, a, \varepsilon\rangle, \langle \perp, \varepsilon\rangle\rangle \mid (A \to a) \in P\}. \end{aligned}$$

□

11. Дополнительные свойства контекстно-свободных языков

11.1*. Деление контекстно-свободных языков

[АхоУль, с. 236], [LewPap2, с. 148–149]

Теорема 11.1. *Пусть L_1 — контекстно-свободный язык над алфавитом Σ и L_2 — автоматный язык над алфавитом Σ. Тогда язык $\{x \in \Sigma^* \mid (\exists y \in L_2)\, xy \in L_1\}$ является контекстно-свободным.*

Доказательство. Пусть $\langle Q_1, \Sigma, \Gamma_1, \Delta_1, I_1, F_1\rangle$ — МП-автомат, распознающий язык L_1. Без ограничения общности можно считать, что для каждого перехода $\langle\langle p, x, \beta\rangle, \langle q, \gamma\rangle\rangle \in \Delta_1$ выполняется неравенство $|x| \leqslant 1$.

Пусть $\langle Q_2, \Sigma, \Delta_2, I_2, F_2\rangle$ — конечный автомат, распознающий язык L_2. Без ограничения общности можно считать, что $Q_1 \cap (Q_1 \times Q_2) = \varnothing$ и для каждого перехода $\langle p, x, q\rangle \in \Delta_2$ выполняется равенство $|x| = 1$.

Тогда язык $\{x \in \Sigma^* \mid (\exists y \in L_2)\, xy \in L_1\}$ распознаётся МП-автоматом $\langle Q, \Sigma, \Gamma_1, \Delta, I, F\rangle$, где $Q = Q_1 \cup (Q_1 \times Q_2)$, $I = I_1$, $F = F_1 \times F_2$ и

$$\begin{aligned}
\Delta = {} & \Delta_1 \cup \\
& \cup \{\langle\langle p, \varepsilon, \varepsilon\rangle, \langle\langle p_1, p_2\rangle, \varepsilon\rangle\rangle \mid p_1 \in Q_1,\ p_2 \in I_2\} \cup \\
& \cup \{\langle\langle\langle p_1, p_2\rangle, \varepsilon, \beta\rangle, \langle\langle q_1, q_2\rangle, \gamma\rangle\rangle \mid \langle\langle p_1, a, \beta\rangle, \langle q_1, \gamma\rangle\rangle \in \Delta_1, \\
& \quad \langle p_2, a, q_2\rangle \in \Delta_2 \text{ для некоторого } a \in \Sigma\} \cup \\
& \cup \{\langle\langle\langle p_1, p_2\rangle, \varepsilon, \beta\rangle, \langle\langle q_1, p_2\rangle, \gamma\rangle\rangle \mid \langle\langle p_1, \varepsilon, \beta\rangle, \langle q_1, \gamma\rangle\rangle \in \Delta_1,\ p_2 \in Q_2\}.
\end{aligned}$$

□

Упражнение 11.2. Существует ли такой контекстно-свободный язык $L \subseteq \Sigma^*$, что язык

$$\mathrm{Subw}(L) \rightleftharpoons \{w \mid w \text{ — подслово некоторого слова из } L\}$$

не является контекстно-свободным?

Ответ. Нет. Положив в теореме 11.1 $L_1 = L$ и $L_2 = \Sigma^*$, получим, что множество всех префиксов слов из языка L является контекстно-свободным. То же верно для множества суффиксов. Далее рассуждаем, как в упражнении 3.7.

11.2. Гомоморфизмы и контекстно-свободные языки

[АхоУль, 2.6.2], [ХопМотУль, 7.3.5], [Гла, 4.2], [LewPap2, с. 148]

Теорема 11.3. *Для любого гомоморфизма $h\colon \Sigma_1^* \to \Sigma_2^*$ и контекстно-свободного языка $L \subseteq \Sigma_1^*$ язык $h(L)$ является контекстно-свободным.*

Доказательство. Приведём доказательство, основанное на МП-автоматах, хотя эту теорему легко доказать и с помощью контекстно-свободных грамматик. Пусть язык L распознаётся МП-автоматом $\langle Q, \Sigma_1, \Gamma, \Delta, I, F\rangle$. Тогда язык $h(L)$ распознаётся МП-автоматом $\langle Q, \Sigma_2, \Gamma, \Delta', I, F\rangle$, где

$$\Delta' = \{\langle\langle p, h(x), \beta\rangle, \langle q, \gamma\rangle\rangle \mid \langle\langle p, x, \beta\rangle, \langle q, \gamma\rangle\rangle \in \Delta\}.$$

□

Пример 11.4. Пусть $\Sigma_1 = \{a, b, c\}$ и $\Sigma_2 = \{a, b\}$. Рассмотрим контекстно-свободный язык L, порождаемый грамматикой $S \to aSSc$, $S \to b$, и гомоморфизм $h\colon \Sigma_1^* \to \Sigma_2^*$, заданный равенствами $h(a) = a$, $h(b) = bba$ и $h(c) = a$. Тогда язык $h(L)$ порождается контекстно-свободной грамматикой $S \to aSSa$, $S \to bba$.

Теорема 11.5. *Для любого гомоморфизма $h\colon \Sigma_1^* \to \Sigma_2^*$ и контекстно-свободного языка $L \subseteq \Sigma_2^*$ язык $h^{-1}(L)$ является контекстно-свободным.*

Доказательство. Обозначим

$$\mathcal{A} = \{x \in \Sigma_2^* \mid x \sqsupset h(a) \text{ для некоторого } a \in \Sigma_1\}.$$

Пусть язык L распознаётся МП-автоматом $\langle Q, \Sigma_2, \Gamma, \Delta, I, F\rangle$. Без ограничения общности можно считать, что для каждого перехода $\langle\langle p, x, \beta\rangle, \langle q, \gamma\rangle\rangle \in \Delta$ выполняется неравенство $|x| \leqslant 1$. Можно проверить, что язык $h^{-1}(L)$ распознаётся МП-автоматом $\langle Q', \Sigma_1, \Gamma, \Delta', I', F'\rangle$, где $Q' = \mathcal{A} \times Q$, $I' = \{\varepsilon\} \times I$, $F' = \{\varepsilon\} \times F$ и

$$\begin{aligned} \Delta' = \{\langle\langle\langle \varepsilon, p\rangle, a, \varepsilon\rangle, \langle\langle h(a), p\rangle, \varepsilon\rangle\rangle \mid a \in \Sigma_1,\ p \in Q\} \cup \\ \cup \{\langle\langle\langle xw, p\rangle, \varepsilon, \beta\rangle, \langle\langle w, q\rangle, \gamma\rangle\rangle \mid \langle\langle p, x, \beta\rangle, \langle q, \gamma\rangle\rangle \in \Delta,\ xw \in \mathcal{A}\}. \end{aligned}$$

$\square$

Пример 11.6. Пусть $\Sigma_1 = \{c, d, e\}$ и $\Sigma_2 = \{a, b\}$. Рассмотрим гомоморфизм $h\colon \Sigma_1^* \to \Sigma_2^*$, заданный равенствами $h(c) = aa$, $h(d) = b$ и $h(e) = bba$. Пусть контекстно-свободный язык L распознаётся МП-автоматом $\langle Q, \Sigma_2, \Gamma, \Delta, I, F\rangle$, где $Q = \{p\}$, $\Gamma = \{A\}$, $\Delta = \{\langle\langle p, a, \varepsilon\rangle, \langle p, A\rangle\rangle,\ \langle\langle p, b, A\rangle, \langle p, \varepsilon\rangle\rangle\}$, $I = \{p\}$ и $F = \{p\}$. Тогда язык $h^{-1}(L)$ распознаётся МП-автоматом $\langle Q', \Sigma_1, \Gamma, \Delta', I', F'\rangle$, где $Q' = \{p_\varepsilon, p_a, p_{aa}, p_b, p_{ba}, p_{bba}\}$, $I' = \{p_\varepsilon\}$, $F' = \{p_\varepsilon\}$ и

$$\begin{aligned} \Delta' = \{&\langle\langle p_\varepsilon, c, \varepsilon\rangle, \langle p_{aa}, \varepsilon\rangle\rangle,\ \langle\langle p_\varepsilon, d, \varepsilon\rangle, \langle p_b, \varepsilon\rangle\rangle,\ \langle\langle p_\varepsilon, e, \varepsilon\rangle, \langle p_{bba}, \varepsilon\rangle\rangle, \\ &\langle\langle p_a, \varepsilon, \varepsilon\rangle, \langle p_\varepsilon, A\rangle\rangle,\ \langle\langle p_{aa}, \varepsilon, \varepsilon\rangle, \langle p_a, A\rangle\rangle,\ \langle\langle p_b, \varepsilon, A\rangle, \langle p_\varepsilon, \varepsilon\rangle\rangle, \\ &\langle\langle p_{ba}, \varepsilon, A\rangle, \langle p_a, \varepsilon\rangle\rangle,\ \langle\langle p_{bba}, \varepsilon, A\rangle, \langle p_{ba}, \varepsilon\rangle\rangle\}. \end{aligned}$$

Язык $h^{-1}(L)$ также порождается контекстно-свободной грамматикой $S \to \varepsilon$, $S \to cTUdS$, $T \to \varepsilon$, $T \to cTUU$, $U \to dT$, $U \to eT$.

11.3*. Представления контекстно-свободных языков посредством гомоморфизмов

[Сал, с. 103–109, 115], [Лал, с. 331–333], [Гла, 6.5], [ГроЛан, 15.2]

Теорема 11.7. *Рассмотрим алфавит* $\Sigma_0 = \{a_1, a_2, b_1, b_2, c\}$ *и язык* $L_0 \subseteq \Sigma_0^*$*, порождаемый контекстно-свободной грамматикой* G_0*:* $S \to Cb_1b_2b_1RC$, $C \to c$, $C \to cDC$, $D \to A$, $D \to AD$, $A \to a_1$, $A \to a_2$, $A \to b_1$, $A \to b_2$, $R \to \varepsilon$, $R \to Ta_1Tb_1$, $R \to Ta_2Tb_2$, $T \to R$, $T \to RCC$.

Произвольный язык $L \subseteq \Sigma^*$ *является контекстно-свободным тогда и только тогда, когда существует такой гомоморфизм* $h\colon \Sigma^* \to \Sigma_0^*$*, что* $L = h^{-1}(L_0)$ *или* $L = h^{-1}(L_0 \cup \{\varepsilon\})$.

Доказательство. Достаточность следует из теоремы 11.5. Приведём теперь идею доказательства необходимости (полное доказательство можно найти в [Сал, с. 103–109]).

Пусть дан произвольный контекстно-свободный язык L. Согласно теореме 8.16 язык $L - \{\varepsilon\}$ порождается некоторой контекстно-свободной грамматикой $\langle\{A_1, \ldots, A_n\}, \Sigma, P, A_1\rangle$, в которой каждое правило имеет один из следующих трёх видов: $A_i \to a$, $A_i \to aA_j$, $A_i \to aA_jA_k$, где $a \in \Sigma$.

Определим вспомогательную функцию $\bar{h}$, ставящую в соответствие каждому символу из Σ конечный язык над алфавитом Σ_0 следующим образом:

$$
\begin{aligned}
\bar{h}(a) = {} & \{b_1b_2^ib_1 \mid (A_i \to a) \in P\} \cup \\
& \cup \{b_1b_2^ib_1a_1a_2^ja_1 \mid (A_i \to aA_j) \in P\} \cup \\
& \cup \{b_1b_2^ib_1a_1a_2^ka_1a_1a_2^ja_1 \mid (A_i \to aA_jA_k) \in P\}.
\end{aligned}
$$

Искомый гомоморфизм h определяется так: если $\bar{h}(a) = \{w_1, w_2, \ldots, w_k\}$, то положим $h(a) = cw_1cw_2c\ldots cw_kc$. □

Пример 11.8. Пусть $\Sigma = \{d, f, g\}$. Рассмотрим язык L, порождаемый грамматикой $A_1 \to f$, $A_1 \to dA_1A_2$, $A_2 \to fA_3$, $A_2 \to fA_2A_3$, $A_3 \to g$. Тогда $L = h^{-1}(L_0)$, где гомоморфизм h задан равенствами

$$
\begin{aligned}
h(d) &= cb_1b_2b_1a_1a_2a_2a_1a_1a_2a_1c,\\
h(f) &= cb_1b_2b_1cb_1b_2b_2b_1a_1a_2a_2a_2a_1cb_1b_2b_2b_1a_1a_2a_2a_2a_1a_1a_2a_2a_1c,\\
h(g) &= cb_1b_2b_2b_2b_1c.
\end{aligned}
$$

Рассмотрим, например, слово $dffg \in L$. Проверим, что слово $h(dffg)$ выводится в грамматике G_0 из теоремы 11.7. Очевидно, $S \overset{*}{\Rightarrow} cb_1b_2b_1Rc$. С помощью последних пяти правил грамматики G_0 можно вывести, что $R \overset{*}{\Rightarrow}$ $\overset{*}{\Rightarrow} a_1a_2^2a_1^2a_2a_1CCb_1b_2b_1CCb_1b_2^2b_1a_1a_2^3a_1CCb_1b_2^3b_1$. Осталось найти такие шесть выводимых из C слов $w_1, \ldots, w_6$, что $h(dffg) =$ $= cb_1b_2b_1a_1a_2^2a_1^2a_2a_1w_1w_2b_1b_2b_1w_3w_4b_1b_2^2b_1a_1a_2^3a_1w_5w_6b_1b_2^3b_1c$. Подходят слова $w_1 = w_2 = w_6 = c$, $w_3 =$ $= cb_1b_2^2b_1a_1a_2^3a_1cb_1b_2^2b_1a_1a_2^3a_1^2a_2^2a_1c$, $w_4 = cb_1b_2b_1c$, $w_5 =$ $= cb_1b_2^2b_1a_1a_2^3a_1^2a_2^2a_1c$.

Теорема 11.9 (Теорема Хомского – Шютценберже). *Произвольный язык $L \subseteq \Sigma^*$ является контекстно-свободным тогда и только тогда, когда существуют натуральное число n, автоматный язык L_1 над алфавитом $\Sigma_n^{(\mathrm{D})} = \{a_1, b_1, a_2, b_2, \ldots, a_n, b_n\}$ и гомоморфизм $h\colon \big(\Sigma_n^{(\mathrm{D})}\big)^* \to \Sigma^*$, такие что $L = h(L_n^{(\mathrm{D})} \cap L_1)$, где $L_n^{(\mathrm{D})}$ — язык Дика над $2n$ буквами.*

Доказательство можно найти в [Лал, с. 331–333].

12. Детерминированные контекстно-свободные языки

12.1. Детерминированные автоматы с магазинной памятью

[АхоУль, 2.5.4], [ХопМотУль, 6.4], [ГорМол, с. 423], [СокКушБад, с. 38], [Рей, с. 88–89]

Определение 12.1. Будем говорить, что два перехода МП-автомата $\langle\langle p_1, x_1, \beta_1\rangle, \langle q_1, \gamma_1\rangle\rangle$ и $\langle\langle p_2, x_2, \beta_2\rangle, \langle q_2, \gamma_2\rangle\rangle$ являются *совместными*, если

1) $p_1 = p_2$,
2) $x_1 \sqsubset x_2$ или $x_2 \sqsubset x_1$,
3) $\beta_1 \sqsubset \beta_2$ или $\beta_2 \sqsubset \beta_1$.

Определение 12.2. МП-автомат называется *детерминированным* (deterministic), если он имеет ровно одно начальное состояние и все переходы этого автомата попарно несовместны.

Пример 12.3. МП-автомат M из примера 10.28 не является детерминированным, так как переходы $\langle\langle 2, a, \varepsilon\rangle, \langle 1, D\rangle\rangle$ и $\langle\langle 2, \varepsilon, D\rangle, \langle 2, \varepsilon\rangle\rangle$ совместны.

Определение 12.4. Язык L над алфавитом Σ называется *детерминированным контекстно-свободным языком*, если существует детерминированный МП-автомат, распознающий язык $L \cdot \{\$\}$ над алфавитом $\Sigma \cup \{\$\}$, где $\$$ — дополнительный символ, не принадлежащий множеству Σ.

Пример 12.5. Пусть $\Sigma = \{a, b\}$. Язык $L = \{a^m b^n \mid m = n \text{ или } n = 0\}$ — детерминированный контекстно-свободный язык, так как язык $L \cdot \{\$\}$ порождается детерминированным МП-автоматом (хотя язык L не порождается никаким детерминированным МП-автоматом).

12.2*. Свойства класса детерминированных контекстно-свободных языков

[АхоУль, 2.5.4, 2.6.4], [Рей, с. 90–93], [ХопМотУль, 6.4], [ГорМол, с. 424–425], [LewPap2, с. 160–161]

Теорема 12.6. *Каждый автоматный язык является детерминированным контекстно-свободным языком.*

Теорема 12.7. *Язык $L \subseteq \Sigma^*$ является детерминированным контекстно-свободным языком тогда и только тогда, когда найдётся такой детерминированный МП-автомат $M' = \langle Q', \Sigma, \Gamma', \Delta', I', F'\rangle$, что*

$$L = \{w \in \Sigma^* \mid \langle s, w, \varepsilon\rangle \vdash^* \langle q, \varepsilon, \alpha\rangle \text{ для некоторых } s \in I',\ q \in F',\ \alpha \in \Gamma'^*\}.$$

Теорема 12.8. *Пусть L — детерминированный контекстно-свободный язык. Тогда язык L не является существенно неоднозначным.*

Теорема 12.9. *Дополнение каждого детерминированного контекстно-свободного языка является детерминированным контекстно-свободным языком.*

Доказательство можно найти в [Гин, с. 110–116] или [АхоУль, с. 212–217].

Пример 12.10. Язык $L = \{a^k b^m c^n \mid k \neq m \text{ или } m \neq n\}$ над алфавитом $\{a, b, c\}$ не является детерминированным контекстно-свободным языком, так как его дополнение не является контекстно-свободным.

Теорема 12.11. *Неверно, что для любых детерминированных контекстно-свободных языков L_1 и L_2 язык $L_1 \cap L_2$ тоже является детерминированным контекстно-свободным языком.*

Доказательство. Достаточно рассмотреть детерминированные контекстно-свободные языки L_1 и L_2 из доказательства теоремы 9.14. □

Теорема 12.12. *Неверно, что для любых детерминированных контекстно-свободных языков L_1 и L_2 язык $L_1 \cup L_2$ тоже является детерминированным контекстно-свободным языком.*

Доказательство. Утверждение следует из теорем 12.9 и 12.11 и закона де Моргана. □

13. Алгоритмические проблемы

13.1. Машины Тьюринга

[АхоУль, 0.4.6], [ХопМотУль, 8.2, 9.2.1], [Гла, 1.4], [Бра, с. 86–91], [Sip, 3.1, 3.2]

Определение 13.1. *Машина Тьюринга* — это семёрка $M = \langle Q, \Sigma, \Gamma, b_0, \Delta, I, F\rangle$, где Q, Γ и Δ — конечные множества, $\Sigma \subset \Gamma$, $b_0 \in \Gamma - \Sigma$, $I \subseteq Q$, $F \subseteq Q$ и $\Delta \subseteq ((Q - F) \times \Gamma) \times (Q \times \Gamma \times \{-1, 0, 1\})$. Здесь Q — *множество состояний*, Σ — *входной алфавит* (*внешний алфавит*), Γ — *ленточный алфавит* (tape alphabet), b_0 — *бланк* (*пробел, пустой символ*) (blank symbol), Δ — *множество переходов*, I — *множество начальных состояний*, F — *множество заключительных состояний*.

Определение 13.2. *Конфигурацией* машины Тьюринга называется любая четвёрка $\langle x, q, a, y\rangle$, где $x \in \Gamma^*$, $q \in Q$, $a \in \Gamma$, $y \in \Gamma^*$.

Определение 13.3. Определим на множестве всех конфигураций машины Тьюринга M бинарное отношение $\underset{M}{\vdash}$ (*такт работы*) следующим образом.

Если $\langle\langle p,a\rangle,\langle q,c,0\rangle\rangle \in \Delta$, то $\langle x,p,a,y\rangle \vdash \langle x,q,c,y\rangle$ для всех $x \in \Gamma^*$ и $y \in \Gamma^*$.

Если $\langle\langle p,a\rangle,\langle q,c,1\rangle\rangle \in \Delta$, то $\langle x,p,a,dy\rangle \vdash \langle xc,q,d,y\rangle$ и $\langle x,p,a,\varepsilon\rangle \vdash \langle xc,q,b_0,\varepsilon\rangle$ для всех $x \in \Gamma^*$, $y \in \Gamma^*$ и $d \in \Gamma$.

Если $\langle\langle p,a\rangle,\langle q,c,-1\rangle\rangle \in \Delta$, то $\langle xd,p,a,y\rangle \vdash \langle x,q,d,cy\rangle$ и $\langle \varepsilon,p,a,y\rangle \vdash \langle \varepsilon,q,b_0,cy\rangle$ для всех $x \in \Gamma^*$, $y \in \Gamma^*$ и $d \in \Gamma$.

Замечание 13.4. Если из контекста ясно, о какой машине Тьюринга идёт речь, вместо $\underset{M}{\vdash}$ будем писать просто $\vdash$.

Определение 13.5. Как и для МП-автомата, для машины Тьюринга бинарное отношение $\overset{*}{\vdash}$ определяется как рефлексивное, транзитивное замыкание отношения $\vdash$.

Замечание 13.6. Если $\langle x_1,q_1,a_1,y_1\rangle \overset{*}{\vdash} \langle x_2,q_2,a_2,y_2\rangle$, то для любых $k \in \mathbb{N}$ и $l \in \mathbb{N}$ найдутся такие $m \in \mathbb{N}$ и $n \in \mathbb{N}$, что $\langle b_0^k x_1,q_1,a_1,y_1 b_0^l\rangle \overset{*}{\vdash} \langle b_0^m x_2,q_2,a_2,y_2 b_0^n\rangle$.

Замечание 13.7. Конфигурацию $\langle b_0^m x,q,a,y b_0^n\rangle$ иногда изображают сокращённо $x\underset{q}{a}y$.

Пример 13.8. Рассмотрим машину Тьюринга $M = \langle Q,\Sigma,\Gamma,b_0,\Delta,I,F\rangle$, где $Q = \{0,1,2,3\}$, $\Sigma = \{a\}$, $\Gamma = \{a,b\}$, $b_0 = b$, $I = \{0\}$, $F = \{3\}$, $\Delta = \{\langle\langle 0,b\rangle,\langle 1,b,1\rangle\rangle, \langle\langle 1,a\rangle,\langle 2,b,1\rangle\rangle, \langle\langle 2,a\rangle,\langle 1,b,1\rangle\rangle, \langle\langle 1,b\rangle,\langle 3,b,0\rangle\rangle\}$. Можно проверить, что для любого $k \in \mathbb{N}$ выполняется следующее: $\underset{0}{b}a^{k+1} \vdash \underset{1}{a}a^k$, $\underset{1}{a}a^{k+2} \overset{*}{\vdash} \underset{1}{a}a^k$, $\underset{1}{a}a^{2k+1} \overset{*}{\vdash} \underset{3}{b}$.

Определение 13.9. Машина Тьюринга $\langle Q,\Sigma,\Gamma,b_0,\Delta,I,F\rangle$ называется *детерминированной*, если множество I содержит ровно один элемент и для каждой пары $\langle p,a\rangle \in (Q-F)\times\Gamma$ существует не более одной тройки $\langle q,c,k\rangle \in Q\times\Gamma\times\{-1,0,1\}$ со свойством $\langle\langle p,a\rangle,\langle q,c,k\rangle\rangle \in \Delta$.

Пример 13.10. Машина Тьюринга из примера 13.8 является детерминированной.

Определение 13.11. Пусть f — частичная функция из Σ^* в Σ^*. Машина Тьюринга $M = \langle Q,\Sigma,\Gamma,b_0,\Delta,I,F\rangle$ *вычисляет* (computes) функцию f тогда и только тогда, когда для каждого слова $w \in \Sigma^*$

1) если $f(w)$ не определено, то не существует таких $p \in I$, $q \in F$, $a \in \Gamma$, $x \in \Gamma^*$, $y \in \Gamma^*$, что $\langle \varepsilon, p, b_0, w\rangle \overset{*}{\vdash} \langle x, q, a, y\rangle$,
2) если $f(w) = z$, то для некоторых $p \in I$, $q \in F$, $m \in \mathbb{N}$ и $n \in \mathbb{N}$ $\langle \varepsilon, p, b_0, w\rangle \overset{*}{\vdash} \langle b_0^m, q, b_0, zb_0^n\rangle$.

Пример 13.12. Машина Тьюринга из примера 13.8 вычисляет следующую частичную функцию:

$$f(a^n) = \begin{cases} \varepsilon, & \text{если число } n \text{ чётно,} \\ \text{не определено,} & \text{если число } n \text{ нечётно.} \end{cases}$$

Пример 13.13*. Рассмотрим детерминированную машину Тьюринга $\langle Q, \Sigma, \Gamma, b_0, \Delta, I, F\rangle$, где $\Sigma = \{a\}$, $\Gamma = \{a, b\}$, $b_0 = b$, $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$, $I = \{1\}$, $F = \{7\}$ и

$$\begin{aligned}
\Delta = \{&\langle\langle 1, b\rangle, \langle 2, b, 1\rangle\rangle,\ \langle\langle 2, b\rangle, \langle 7, b, 0\rangle\rangle,\ \langle\langle 2, a\rangle, \langle 3, a, -1\rangle\rangle,\\
&\langle\langle 3, b\rangle, \langle 3, a, 1\rangle\rangle,\ \langle\langle 3, a\rangle, \langle 4, b, 1\rangle\rangle,\\
&\langle\langle 4, b\rangle, \langle 5, b, -1\rangle\rangle,\ \langle\langle 4, a\rangle, \langle 8, a, 0\rangle\rangle,\\
&\langle\langle 5, b\rangle, \langle 6, b, -1\rangle\rangle,\ \langle\langle 6, a\rangle, \langle 6, a, -1\rangle\rangle,\ \langle\langle 6, b\rangle, \langle 7, b, 0\rangle\rangle,\\
&\langle\langle 8, b\rangle, \langle 8, b, 1\rangle\rangle,\ \langle\langle 8, a\rangle, \langle 9, a, 1\rangle\rangle,\\
&\langle\langle 9, a\rangle, \langle 9, a, 1\rangle\rangle,\ \langle\langle 9, b\rangle, \langle 10, b, -1\rangle\rangle,\\
&\langle\langle 10, a\rangle, \langle 11, b, -1\rangle\rangle,\ \langle\langle 11, b\rangle, \langle 11, b, 0\rangle\rangle,\ \langle\langle 11, a\rangle, \langle 12, b, -1\rangle\rangle,\\
&\langle\langle 12, a\rangle, \langle 12, a, -1\rangle\rangle,\ \langle\langle 12, b\rangle, \langle 13, b, -1\rangle\rangle,\\
&\langle\langle 13, b\rangle, \langle 13, b, -1\rangle\rangle,\ \langle\langle 13, a\rangle, \langle 14, b, -1\rangle\rangle,\\
&\langle\langle 14, a\rangle, \langle 8, a, 1\rangle\rangle,\ \langle\langle 14, b\rangle, \langle 3, b, 1\rangle\rangle\}.
\end{aligned}$$

Можно проверить, что для любых $i \in \mathbb{N}$, $j \in \mathbb{N}$ и $k \in \mathbb{N}$ выполняется следующее: $\underset{1}{b}a^{k+2} \overset{*}{\vdash} ab\underset{8}{a}a^k$, $ab^{j+1}\underset{8}{a}a^2 \overset{*}{\vdash} \underset{7}{b}a^{j+2}$, $ab^{j+1}\underset{8}{a}a^{k+3} \overset{*}{\vdash} a^{j+2}b\underset{8}{a}a^k$, $a^{i+2}b^{j+1}\underset{8}{a}a^{k+2} \overset{*}{\vdash} a^{i+1}b^{j+2}\underset{8}{a}a^k$, $ab^{j+1}\underset{8}{a}a^{k+2j+5} \overset{*}{\vdash} ab^{j+2}\underset{8}{a}a^k$. Следовательно, $\underset{1}{b}a^{i+(j+2)^2} \overset{*}{\vdash} ab^{j+1}\underset{8}{a}a^{i+2}$ и $\underset{1}{b}a^{(k+2)^2} \overset{*}{\vdash} \underset{7}{b}a^{k+2}$. Рассматриваемая машина Тьюринга вычисляет следующую частичную функцию:

$$f(a^n) = \begin{cases} a^{\sqrt{n}}, & \text{если } \sqrt{n} \in \mathbb{N}, \\ \text{не определено,} & \text{если } \sqrt{n} \notin \mathbb{N}. \end{cases}$$

Определение 13.14. Говорят, что детерминированная машина Тьюринга $\langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_{\mathrm{s}}\}, \{q_{\mathrm{a}}, q_{\mathrm{r}}\}\rangle$ с выделенным состоянием q_{a} *разрешает* (decides) язык $L \subseteq \Sigma^*$, если

1) для каждого слова $w \in L$ найдутся такие $m \in \mathbb{N}$ и $n \in \mathbb{N}$, что $\langle \varepsilon, q_{\mathrm{s}}, b_0, w\rangle \overset{*}{\vdash} \langle b_0^m, q_{\mathrm{a}}, b_0, b_0^n\rangle$,
2) для каждого слова $w \in \Sigma^* - L$ найдутся такие $m \in \mathbb{N}$ и $n \in \mathbb{N}$, что $\langle \varepsilon, q_{\mathrm{s}}, b_0, w\rangle \overset{*}{\vdash} \langle b_0^m, q_{\mathrm{r}}, b_0, b_0^n\rangle$.

Состояние q_{a} называется *допускающим* (accept state), состояние q_{r} называется *отвергающим* (reject state).

Пример 13.15. Рассмотрим детерминированную машину Тьюринга $M = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_{\mathrm{s}}\}, \{q_{\mathrm{a}}, q_{\mathrm{r}}\}\rangle$, где $Q = \{0, 1, 2, 3, 4, 5\}$, $\Sigma = \{a\}$, $\Gamma = \{a, b\}$, $b_0 = b$, $q_{\mathrm{s}} = 0$, $q_{\mathrm{a}} = 4$, $q_{\mathrm{r}} = 5$ и

$$\begin{aligned}\Delta = \{&\langle\langle 0, b\rangle, \langle 1, b, 1\rangle\rangle,\\ &\langle\langle 1, a\rangle, \langle 2, b, 1\rangle\rangle,\ \langle\langle 2, a\rangle, \langle 3, b, 1\rangle\rangle,\ \langle\langle 3, a\rangle, \langle 1, b, 1\rangle\rangle,\\ &\langle\langle 1, b\rangle, \langle 4, b, 0\rangle\rangle,\ \langle\langle 2, b\rangle, \langle 5, b, 0\rangle\rangle,\ \langle\langle 3, b\rangle, \langle 5, b, 0\rangle\rangle\}.\end{aligned}$$

Эта машина Тьюринга разрешает язык $\{a^{3n} \mid n \in \mathbb{N}\}$.

Определение 13.16. Язык L над алфавитом Σ называется *разрешимым* или *рекурсивным* (decidable, recursive), если существует детерминированная машина Тьюринга $\langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_{\mathrm{s}}\}, \{q_{\mathrm{a}}, q_{\mathrm{r}}\}\rangle$ (с выделенным состоянием q_{a}), которая разрешает язык L.

Определение 13.17. Говорят, что машина Тьюринга $M = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_{\mathrm{s}}\}, \{q_{\mathrm{a}}\}\rangle$ *допускает* (accepts) слово $w \in \Sigma^*$, если для некоторых $m \in \mathbb{N}$ и $n \in \mathbb{N}$ $\langle \varepsilon, q_{\mathrm{s}}, b_0, w\rangle \overset{*}{\vdash} \langle b_0^m, q_{\mathrm{a}}, b_0, b_0^n\rangle$.

Определение 13.18. Язык, *допускаемый* машиной Тьюринга M, — это язык, состоящий из всех допускаемых данной машиной Тьюринга слов.

Определение 13.19. Язык называется *перечислимым* (или *рекурсивно перечислимым*, или *полуразрешимым*) (recursively enumerable), если существует детерминированная машина Тьюринга, допускающая этот язык.

Замечание 13.20. В определении 13.19 можно отбросить требование детерминированности машины Тьюринга.

Теорема 13.21. *Каждый разрешимый язык является перечислимым.*

Доказательство. Пусть дана машина Тьюринга $M = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_\mathrm{s}\}, \{q_\mathrm{a}, q_\mathrm{r}\}\rangle$ с выделенным состоянием q_a, которая разрешает язык $L \subseteq \Sigma^*$. Тогда машина Тьюринга $M' = \langle Q, \Sigma, \Gamma, b_0, \Delta, \{q_\mathrm{s}\}, \{q_\mathrm{a}\}\rangle$ допускает язык L. □

Пример 13.22*. Если в машине Тьюринга из примера 13.13* заменить переход $\langle\langle 6, a\rangle, \langle 6, a, -1\rangle\rangle$ на $\langle\langle 6, a\rangle, \langle 6, b, -1\rangle\rangle$, то получится машина Тьюринга, допускающая язык $\{a^{n^2} \mid n \in \mathbb{N}\}$. Следовательно, этот язык является перечислимым. Можно доказать, что он даже является разрешимым.

13.2. Массовые задачи

[АхоУль, 0.4.5], [Гла, 8.1], [Sip, 4.2], [ХопМотУль, 9.2]

Определение 13.23. *Массовой задачей* (problem) называется бесконечная серия "однотипных" *индивидуальных задач* (instance), каждая из которых имеет определённый *ответ*. С каждой массовой задачей связана некоторая фиксированная *схема кодирования* (encoding scheme), которая отображает индивидуальные задачи в их коды — слова в некотором фиксированном алфавите. При этом требуется, чтобы множество кодов всех индивидуальных задач было разрешимым.

Определение 13.24. *Задачей распознавания* (decision problem) называется массовая задача, в которой ответами индивидуальных задач могут быть только "да" и "нет" (то есть существует только два возможных ответа).

Пример 13.25. Зафиксируем некоторый алфавит Σ. Рассмотрим следующие однотипные индивидуальные задачи: даны два слова $x \in \Sigma^*$ и $y \in \Sigma^*$, необходимо выяснить, является ли слово x подсловом слова y.

Пусть $\#$ — новый символ, не принадлежащий алфавиту Σ. Кодом индивидуальной задачи про конкретные слова x и y будем считать слово $x\#y$ в алфавите $\Sigma \cup \{\#\}$.

Эта массовая задача является задачей распознавания.

Определение 13.26. *Алгоритмическая проблема* — проблема, в которой требуется найти алгоритм для решения массовой задачи. Если такой алгоритм существует, то данная проблема называется *алгоритмически разрешимой* или просто *разрешимой* (decidable, solvable), иначе — *алгоритмически неразрешимой* (undecidable, unsolvable).

Замечание 13.27. Алгоритмическая проблема, относящаяся к некоторой задаче распознавания, алгоритмически разрешима тогда и только тогда, когда разрешимо множество кодов индивидуальных задач с ответом "да".

Пример 13.28. Рассмотрим массовую задачу из примера 13.25. Соответствующая алгоритмическая проблема разрешима, так как разрешим язык $\{x\#uxv \mid x \in \Sigma^*,\ u \in \Sigma^*,\ v \in \Sigma^*\}$.

Пример 13.29. Зафиксируем некоторый алфавит Σ. Рассмотрим следующие однотипные индивидуальные задачи: дана порождающая грамматика $G = \langle\{A_1, \ldots, A_n\}, \Sigma, P, A_1\rangle$, необходимо выяснить, является ли эта грамматика праволинейной.

Для полной строгости необходимо договориться, как кодировать грамматику G в виде слова. Например, можно использовать алфавит $\Sigma \cup \{\heartsuit, \spadesuit, 0, 1\}$, где $\heartsuit$, $\spadesuit$, 0, 1 — дополнительные символы, не принадлежащие множеству Σ. Вспомогательный символ A_i заменим на слово, состоящее из символа $\heartsuit$ и кода числа i в двоичной системе счисления. В каждом правиле $\alpha \to \beta$ добавим символ $\spadesuit$ на месте $\to$ и после слова β. Кодом грамматики G будем считать результат конкатенации кодов всех правил из множества P. Например, грамматика $A_1 \to a$, $A_1 \to A_2A_1$, $A_2 \to aA_3$, $A_3 \to \varepsilon$, $A_3 \to bA_1$ (над алфавитом $\Sigma = \{a, b\}$) кодируется словом $\heartsuit 1 \spadesuit a \spadesuit \heartsuit 1 \spadesuit \heartsuit 10 \heartsuit 1 \spadesuit \heartsuit 10 \spadesuit a \heartsuit 11 \spadesuit \heartsuit 11 \spadesuit \spadesuit \heartsuit 11 \spadesuit b \heartsuit 1 \spadesuit$.

Легко понять, что соответствующая алгоритмическая проблема (проблема проверки праволинейности) разрешима.

Теорема 13.30. *Существует такая машина Тьюринга над однобуквенным алфавитом $\Sigma = \{a\}$, что язык, допускаемый этой машиной Тьюринга, неразрешим.*

Доказательство. Доказательство существования перечислимого неразрешимого множества можно найти, например, в [ХопМотУль, 9.2]. □

13.3*. Грамматики типа 0

[ЛПИИ, 5.2.8], [Гла, 1.4], [АхоУль, с. 50, 120]

Теорема 13.31. *Класс языков, порождаемых грамматиками типа 0, совпадает с классом перечислимых языков.*

Упражнение 13.32. Выводимо ли слово aab в грамматике $S \to Sb$, $S \to ST$, $S \to SU$, $S \to \varepsilon$, $Ta \to aaT$, $Tb \to ab$, $Uaaa \to aaU$, $Uab \to b$?
Ответ. Да. $S \overset{*}{\Rightarrow} UUTTTb \overset{*}{\Rightarrow} UUaaaaaaab \overset{*}{\Rightarrow} aab$.

Замечание 13.33. Неизвестно, порождает ли грамматика $R \to Ra$, $R \to S$, $S \to Sb$, $S \to ST$, $S \to SU$, $S \to \varepsilon$, $Ta \to aaT$, $Tb \to ab$, $Uaaa \to aaU$, $Uab \to b$ все слова в алфавите $\{a, b\}$.

Упражнение 13.34. Пусть $\Sigma = \{a, b, c, d, e\}$. Рассмотрим грамматику $R \to ccdcdcddccddcdccca$, $ac \to ca$, $ca \to ac$, $bc \to cb$, $cb \to bc$, $ad \to da$, $da \to ad$, $bd \to db$, $db \to bd$, $ce \to eca$, $eca \to ce$, $de \to edb$, $edb \to de$, $cca \to ccae$, $ccae \to cca$. Выводимо ли в ней слово $ccdcdcddccddcdcccabba$?
Ответ. Да. Пусть $w = ccdcdcddccddcdccc$. Тогда $wa \overset{*}{\Rightarrow}$ $\overset{*}{\Rightarrow} wabababba \overset{*}{\Rightarrow} wabbababababba \overset{*}{\Rightarrow} wabbabababababbababba \overset{*}{\Rightarrow}$ $\overset{*}{\Rightarrow} wabbababababba \overset{*}{\Rightarrow} wabba$.

Упражнение 13.35. Является ли разрешимым язык, порождаемый грамматикой $S \to cS$, $S \to dS$, $S \to aa$, $ac \to ca$, $ca \to ac$, $bc \to cb$, $cb \to bc$, $ad \to da$, $da \to ad$, $bd \to db$, $db \to bd$, $cdc \to cdcR$, $Rcca \to ccR$, $Rddb \to ddR$, $Rcdd \to cddT$, $Tcc \to ccaT$, $Tdd \to ddbT$, $Tcdc \to cdc$?
Ответ. Нет. Неразрешимо пересечение этого языка с языком $cdc((cc+dd)^*cdd(cc+dd)^*cdc)^*(a+b)^*$. В пересечении можно закодировать любую грамматику типа 0 (вспомогательный символ A_i заменяется на $ccd^{2i}cc$, в каждом правиле $\alpha \to \beta$ добавляется слово cdd на месте $\to$ и слово cdc после слова β).

Определение 13.36. *Порождающая грамматика в нормальной форме* — порождающая грамматика, в которой каждое правило имеет вид $A \to a$, $A \to BC$ или $AB \to \varepsilon$, где $A \in N$, $B \in N$, $C \in N$, $a \in \Sigma$.

Теорема 13.37. *Каждая порождающая грамматика эквивалентна некоторой порождающей грамматике в нормальной форме.*

13.4. Проблема соответствий Поста

[Sip, 5.2], [Гин, 4.2], [Сал, 4.6], [АхоУль, 0.4.6], [ГорМол, с. 375], [ХопМотУль, 9.4], [ГроЛан, 6.2.3]

Определение 13.38. *Постовской системой соответствия* над алфавитом Σ называется пара конечных последовательностей $((x_1,\ldots,x_n),(y_1,\ldots,y_n))$, где $x_i\in\Sigma^*$ и $y_i\in\Sigma^*$ для всех i.

Замечание 13.39. Систему $((x_1,\ldots,x_n),(y_1,\ldots,y_n))$ иногда изображают в виде $\left[\dfrac{x_1}{y_1}\right],\ldots,\left[\dfrac{x_n}{y_n}\right]$.

Определение 13.40. *Решением* (match) постовской системы соответствия $((x_1,\ldots,x_n),(y_1,\ldots,y_n))$ называется непустая последовательность индексов $(i_1,\ldots,i_k)$, удовлетворяющая условию $x_{i_1}\ldots x_{i_k}=y_{i_1}\ldots y_{i_k}$, где $1\leqslant i_j\leqslant n$ для каждого j.

Пример 13.41. Пусть $\Sigma=\{a,b,c\}$. Рассмотрим постовскую систему соответствия $\left[\dfrac{aab}{a}\right],\left[\dfrac{a}{aa}\right],\left[\dfrac{caa}{bc}\right]$. Последовательность $(2,1,3,2,2)$ является решением этой системы.

Упражнение 13.42. Пусть $\Sigma=\{a,b,c,0,1,\diamondsuit,\heartsuit,\spadesuit\}$. Существует ли решение у постовской системы соответствия

$$\left[\frac{a\spadesuit}{a\spadesuit aba\heartsuit 1aca\spadesuit a}\right],\left[\frac{ab}{ba}\right],\left[\frac{ac}{ca}\right],\left[\frac{a\spadesuit}{\spadesuit a}\right],\left[\frac{a\spadesuit}{ba\spadesuit a}\right],\left[\frac{a\spadesuit}{\spadesuit aba}\right],$$
$$\left[\frac{a\heartsuit 1ab}{\diamondsuit a}\right],\left[\frac{a\heartsuit 1ac}{ca\heartsuit 10a}\right],\left[\frac{aba\heartsuit 10ab}{\heartsuit 10abaca}\right],\left[\frac{aca\heartsuit 10ab}{\heartsuit 10acaca}\right],\left[\frac{a\heartsuit 10ac}{ca\heartsuit 11a}\right],\left[\frac{a\heartsuit 11ac}{ca\heartsuit 1a}\right],$$
$$\left[\frac{a\diamondsuit ab}{\diamondsuit a}\right],\left[\frac{a\diamondsuit ac}{\diamondsuit a}\right],\left[\frac{aba\diamondsuit}{\diamondsuit a}\right],\left[\frac{aca\diamondsuit}{\diamondsuit a}\right],\left[\frac{a\diamondsuit a\spadesuit a\spadesuit}{\spadesuit}\right]?$$

Ответ. Да. Например, (1, 2, 8, 5, 2, 10, 4, 2, 11, 3, 4, 2, 3, 12, 5, 2, 3, 3, 7, 4, 2, 3, 16, 4, 2, 16, 4, 15, 4, 17).

Определение 13.43. *Проблемой соответствий Поста* (Post correspondence problem) называется проблема нахождения алгоритма, выясняющего для каждой постовской системы соответствия, существует ли решение этой системы.

Теорема 13.44. *Пусть $|\Sigma|\geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольной постовской системе соответствия над алфавитом Σ узнать, имеет ли она решение. (То есть проблема соответствий Поста неразрешима.)*

Доказательство можно найти в [ХопМотУль, 9.4].

14. Алгоритмически разрешимые проблемы

14.1. Неукорачивающие грамматики

[ГроЛан, 12.1], [АхоУль, с. 117, 119], [Бра, с. 37–38], [Гла, 3.2, 3.5], [Лал, с. 161], [ГлаМел, с. 54], [ГорМол, с. 361–362], [LewPap2, с. 271], [КукБей, с. 271–272], [ЛПИИ, 5.2.7]

Определение 14.1. Порождающая грамматика называется *неукорачивающей*, если для каждого правила $(\alpha \to \beta) \in P$ выполняется неравенство $|\alpha| \leqslant |\beta|$.

Теорема 14.2. *Существует алгоритм, позволяющий по произвольной неукорачивающей грамматике G и по слову w узнать, верно ли, что $w \in L(G)$.*

Теорема 14.3. *Каждая контекстная грамматика является неукорачивающей. Каждая неукорачивающая грамматика эквивалентна некоторой контекстной грамматике.*

Пример 14.4. Грамматика $S \to ASTA$, $S \to AbA$, $A \to a$, $bT \to bb$, $AT \to TA$ эквивалентна контекстной грамматике из примера 1.59.

14.2*. Линейно ограниченные автоматы

[Бра, с. 91–100], [Гин, с. 107–108], [Гла, 3.2], [АхоУль, с. 120], [ГроЛан, 12.2], [Sip, с. 177], [LewPap2, с. 271]

Определение 14.5. Машина Тьюринга $\langle Q, \Sigma, \Gamma, b_0, \Delta, I, F\rangle$ называется *линейно ограниченным автоматом* (linear bounded automaton), если не существует таких $w \in \Sigma^*$, $x \in \Gamma^*$, $q \in Q$, $a \in \Gamma$ и $y \in \Gamma^*$, что $\langle \varepsilon, p, b_0, wb_0\rangle \overset{*}{\vdash} \langle x, q, a, y\rangle$ и $|xay| > |b_0 w b_0|$.

Теорема 14.6. *Язык L, не содержащий пустого слова, порождается некоторой неукорачивающей грамматикой тогда и только тогда, когда существует линейно ограниченный автомат (в общем случае недетерминированный), который допускает язык L.*

Замечание 14.7. Неизвестно, верен ли аналог теоремы 14.6 для детерминированных линейно ограниченных автоматов.

Теорема 14.8. *Класс языков, порождаемых неукорачивающими грамматиками, замкнут относительно операций объединения, пересечения и дополнения.*

14.3. Проблема выводимости слова

[Гин, 4.1, 1.8], [ХопМотУль, 7.4.4], [Лал, с. 305], [ГроЛан, 7.2.4], [Sip, 4.1], [LewPap2, 3.6]

Теорема 14.9. *Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике G узнать, верно ли, что $\varepsilon \in L(G)$.*

Теорема 14.10. *Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике G и по слову w узнать, верно ли, что $w \in L(G)$.*

14.4. Проблема пустоты языка

[Гин, 4.1, 1.4], [АхоУль, 2.4.2], [ХопМотУль, 7.4.3], [Гла, 4.2], [ГроЛан, 7.2.3], [Лал, с. 305], [Sip, 4.1], [Рей, с. 65–66]

Теорема 14.11. *Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике G узнать, верно ли, что $L(G) = \varnothing$.*

Доказательство. Удалим из G все бесполезные символы, кроме начального символа (как в доказательстве теоремы 8.2). Если полученная грамматика содержит хотя бы одно правило, то $L(G) \neq \varnothing$. □

14.5. Проблема бесконечности языка

[Гин, 4.1], [ГроЛан, 7.2.3], [Гла, 4.2], [Лал, с. 297–299, 305], [Бра, с. 65–70], [Рей, с. 71]

Теорема 14.12. *Существует алгоритм, позволяющий по произвольной контекстно-свободной грамматике G узнать, является ли бесконечным язык $L(G)$.*

Пример 14.13. Рассмотрим контекстно-свободную грамматику G из примера 8.3. Чтобы выяснить, является ли язык $L(G)$ бесконечным, удалим сначала все бесполезные символы. Затем устраним все ε-правила. Так как $W \overset{*}{\Rightarrow} Z$ и $Z \overset{*}{\Rightarrow} W$, то можно всюду заменить W на Z. Получится грамматика $S \to VZ$, $T \to aa$, $T \to bb$, $V \to aTb$, $V \to bTa$, $Z \to aab$, $Z \to b$, эквивалентная исходной грамматике G. Очевидно, эта грамматика не содержит "рекурсивных" нетерминальных символов. Следовательно, язык $L(G)$ является конечным.

14.6. Проблема равенства автоматных языков

[Гин, 4.1], [АхоУль, 2.3.4], [Сал, с. 35], [ХопМотУль, 4.4.2], [ГроЛан, 10.4.5], [ТраБар, 1.1, 1.4], [Sip, 4.1], [LewPap2, 2.6], [Рей, с. 62–63]

Теорема 14.14. *Существует алгоритм, позволяющий по произвольным двум праволинейным грамматикам G_1 и G_2 узнать, верно ли, что $L(G_1) \subseteq L(G_2)$.*

Теорема 14.15. *Существует алгоритм, позволяющий по произвольным двум праволинейным грамматикам G_1 и G_2 узнать, верно ли, что $L(G_1) = L(G_2)$.*

15. Алгоритмически неразрешимые проблемы

15.1. Пересечение контекстно-свободных языков

[Гин, 4.2], [ГроЛан, 8.1.1, 8.1.2], [АхоУль, с. 237], [Лал, с. 306]

Определение 15.1. Пусть $\vec{x} = (x_1, \ldots, x_n)$, где $x_i \in \{a, b\}^*$ для всех i. Рассмотрим алфавит $\Sigma_3 = \{a, b, c\}$. Обозначим через $G(\vec{x})$ линейную грамматику $\langle \{S\}, \Sigma_3, P, S \rangle$, где $P = \{S \to ba^i S x_i \mid 1 \leqslant i \leqslant n\} \cup \{S \to ba^i c x_i \mid 1 \leqslant i \leqslant n\}$. Обозначим через $\mathcal{L}_{\vec{x}}$ язык, порождаемый грамматикой $G(\vec{x})$.

Лемма 15.2. *Язык $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}}$ является непустым тогда и только тогда, когда постовская система соответствия $(\vec{x}, \vec{y})$ имеет решение.*

Пример 15.3. Рассмотрим постовскую систему соответствия

$$\left[\frac{b}{bba}\right], \left[\frac{abbaa}{a}\right]$$

(то есть $n = 2$, $\vec{x} = (b, abbaa)$ и $\vec{y} = (bba, a)$). Решениями этой системы являются последовательности $(1, 1, 2)$, $(1, 1, 2, 1, 1, 2)$ и т. д. Легко убедиться, что $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}} = \{(baababa)^n c(bbabbaa)^n \mid n \geqslant 1\}$.

Теорема 15.4. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам G_1 и G_2 над алфавитом Σ узнать, верно ли, что $L(G_1) \cap L(G_2) = \varnothing$.*

Доказательство. Сначала докажем утверждение теоремы для случая $|\Sigma| \geqslant 3$. Из леммы 15.2 следует, что если бы проблема распознавания свойства $L(G_1) \cap L(G_2) = \varnothing$ для контекстно-свободных грамматик над алфавитом Σ была разрешима, то проблема соответствий Поста тоже была бы разрешима. Поэтому из неразрешимости проблемы соответствий Поста следует неразрешимость проблемы распознавания свойства $L(G_1) \cap L(G_2) = \varnothing$ для контекстно-свободных грамматик над алфавитом Σ.

Чтобы доказать утверждение теоремы для случая $|\Sigma| = 2$ (например, $\Sigma = \{d, e\}$), достаточно заменить в определении $G(\vec{x})$ символ a на ede, символ b на $edde$ и символ c на $eddde$. □

Лемма 15.5. *Язык $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}}$ является бесконечным тогда и только тогда, когда постовская система соответствия $(\vec{x}, \vec{y})$ имеет решение.*

Доказательство. Если постовская система соответствия имеет хотя бы одно решение, то она имеет бесконечно много решений. □

Теорема 15.6. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам G_1 и G_2 над алфавитом Σ узнать, является ли бесконечным язык $L(G_1) \cap L(G_2)$.*

15.2. Проблема однозначности

[Гин, 4.5], [АхоУль, 2.6.5], [ГроЛан, 8.2.4], [Лал, с. 307], [ХопМотУль, 9.5.2], [Гла, 8.4]

Теорема 15.7. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике G над алфавитом Σ узнать, является ли грамматика G однозначной.*

Доказательство. Рассмотрим язык $\mathcal{L}_{\vec{x}} \cup \mathcal{L}_{\vec{y}}$. Следуя доказательству теоремы 9.12, построим грамматику G для этого языка, исходя из грамматик $G(\vec{x})$ и $G(\vec{y})$. Грамматика G является неоднозначной тогда и только тогда, когда постовская система соответствия $(\vec{x}, \vec{y})$ имеет решение. □

15.3. Дополнение контекстно-свободного языка

[Гин, 4.2], [ГроЛан, 8.1.4], [АхоУль, 2.6.3], [ХопМотУль, 9.5.3], [Sip, с. 181–182], [Сал, 5.3], [LewPap2, 5.5], [Гла, 8.4]

Лемма 15.8. *Рассмотрим алфавит $\Sigma_3 = \{a, b, c\}$. Язык $\Sigma_3^* - \mathcal{L}_{\vec{x}}$ является контекстно-свободным.*

Пример 15.9. Пусть $\vec{x} = (b, abbaa)$. Тогда язык $\Sigma_3^* - \mathcal{L}_{\vec{x}}$ над алфавитом $\Sigma_3 = \{a, b, c\}$ порождается контекстно-свободной грамматикой $S \to \varepsilon$, $S \to AW$, $S \to bR$, $A \to a$, $A \to c$, $B \to b$, $B \to c$, $Z \to a$, $Z \to B$, $W \to ZW$, $W \to \varepsilon$, $U_a \to WB$, $U_a \to \varepsilon$, $U_b \to WA$, $U_b \to \varepsilon$, $R \to \varepsilon$, $R \to BW$, $R \to aaaW$, $R \to a$, $R \to aa$, $R \to abRb$, $R \to aabRabbaa$, $R \to acZWb$, $R \to aacZWabbaa$, $R \to aBT_1$, $R \to aaBT_2$, $T_1 \to U_b$, $T_2 \to U_abbaa$, $T_2 \to U_bbaa$, $T_2 \to U_baa$, $T_2 \to U_aa$, $T_2 \to U_a$. Заметим, что $\{w \in \Sigma_3^* \mid T_i \overset{*}{\Rightarrow} w\} = \Sigma_3^* - (\Sigma_3^* \cdot \{x_i\})$ для каждого i.

Язык $\Sigma_3^* - \mathcal{L}_{\vec{x}}$ является даже линейным (чтобы получить линейную грамматику, достаточно “раскрыть” вспомогательные символы A, B и Z).

Замечание 15.10. Лемму 15.8 можно доказать, явно построив контекстно-свободную грамматику (как в примере 15.9), а можно и вывести из теоремы 12.9, проверив, что $\mathcal{L}_{\vec{x}}$ — детерминированный контекстно-свободный язык.

Определение 15.11. Пусть $\Sigma_3 = \{a, b, c\}$, $\vec{x} = (x_1, \ldots, x_n)$, $\vec{y} = (y_1, \ldots, y_n)$, где $x_i \in \{a, b\}^*$ и $y_i \in \{a, b\}^*$ для всех i. Обозначим через $\mathcal{M}_{\vec{x},\vec{y}}$ язык $\Sigma_3^* - (\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}})$.

Лемма 15.12. *Язык $\mathcal{M}_{\vec{x},\vec{y}}$ является контекстно-свободным.*

Доказательство. $\mathcal{M}_{\vec{x},\vec{y}} = (\Sigma_3^* - \mathcal{L}_{\vec{x}}) \cup (\Sigma_3^* - \mathcal{L}_{\vec{y}})$. □

Лемма 15.13. *Дополнение языка $\mathcal{M}_{\vec{x},\vec{y}}$ является непустым тогда и только тогда, когда постовская система соответствия $(\vec{x}, \vec{y})$ имеет решение.*

Доказательство. Утверждение следует из леммы 15.2. □

Теорема 15.14. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике G над алфавитом Σ узнать, верно ли, что $L(G) = \Sigma^*$.*

Доказательство. Очевидно, $L(G) = \Sigma^*$ тогда и только тогда, когда дополнение языка $L(G)$ является пустым. □

Теорема 15.15. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам G_1 и G_2 над алфавитом Σ узнать, верно ли, что $L(G_1) = L(G_2)$.*

Доказательство. Утверждение следует из предыдущей теоремы и примера 1.71. □

Теорема 15.16. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам G_1 и G_2 над алфавитом Σ узнать, верно ли, что $L(G_1) \subseteq L(G_2)$.*

Доказательство. Очевидно, $\Sigma^* \subseteq L(G)$ тогда и только тогда, когда $L(G) = \Sigma^*$. □

Лемма 15.17. *Дополнение языка $\mathcal{M}_{\vec{x},\vec{y}}$ является бесконечным тогда и только тогда, когда постовская система соответствия $(\vec{x}, \vec{y})$ имеет решение.*

Теорема 15.18. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике G над алфавитом Σ узнать, является ли бесконечным множество $\Sigma^* - L(G)$.*

15.4. Проблема автоматности

[Гин, 4.2], [ГроЛан, 8.1.4], [Лал, с. 306], [АхоУль, с. 237], [ХопМотУль, 9.5.3], [Гла, 8.4]

Лемма 15.19. *Пусть* $\vec{x} = (x_1, \ldots, x_n)$, $\vec{y} = (y_1, \ldots, y_n)$, *где* $x_i \in \{a,b\}^*$, $y_i \in \{a,b\}^*$ *и* $x_i y_i \neq \varepsilon$ *для всех* i. *Тогда язык* $\mathcal{M}_{\vec{x},\vec{y}}$ *является автоматным тогда и только тогда, когда постовская система соответствия* $(\vec{x}, \vec{y})$ *не имеет решений.*

Доказательство. Пусть $(i_1, \ldots, i_k)$ — решение постовской системы соответствия $(\vec{x}, \vec{y})$, где $x_i y_i \neq \varepsilon$ для всех i. Обозначим $u \rightleftharpoons ba^{i_k} ba^{i_{k-1}} \ldots ba^{i_1}$, $v \rightleftharpoons x_{i_1} \ldots x_{i_k}$ и $L_0 \rightleftharpoons \{u\}^* \cdot \{c\} \cdot \{v\}^*$. Легко проверить, что $u \neq \varepsilon$, $v \neq \varepsilon$ и язык L_0 является автоматным. Очевидно, $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}} \cap L_0 = \{u^m c v^m \mid m > 0\}$. Как было установлено в упражнении 3.13, язык $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}} \cap L_0$ не является автоматным. Согласно теореме 3.8 язык $\mathcal{L}_{\vec{x}} \cap \mathcal{L}_{\vec{y}}$ не является автоматным. Следовательно, и язык $\mathcal{M}_{\vec{x},\vec{y}}$ не является автоматным.

Обратно, если постовская система соответствия $(\vec{x}, \vec{y})$ не имеет решений, то $\mathcal{M}_{\vec{x},\vec{y}} = \{a,b,c\}^*$, а этот язык является автоматным. □

Теорема 15.20. *Пусть* $|\Sigma| \geqslant 2$. *Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике* G *над алфавитом* Σ *узнать, является ли автоматным язык* $L(G)$.

Доказательство. Докажем утверждение теоремы для случая $|\Sigma| \geqslant 3$. Из леммы 15.19 следует, что если бы проблема распознавания автоматности языка $L(G)$ для контекстно-свободных грамматик над алфавитом Σ была разрешима, то также была бы разрешима проблема соответствий Поста для постовских систем соответствия $((x_1, \ldots, x_n), (y_1, \ldots, y_n))$, где $x_i \in \{a,b\}^*$, $y_i \in \{a,b\}^*$ и $x_i y_i \neq \varepsilon$ для всех i. Но тогда была бы разрешима проблема соответствий Поста для всех постовских систем соответствия над алфавитом $\{a,b\}$ (если $x_i y_i = \varepsilon$ для некоторого i, то рассматриваемая постовская система соответствия имеет решение, а именно (i)). □

15.5. Проблемы контекстно-свободности

[Гин, 4.2], [ГроЛан, 8.1.3, 8.1.4], [Лал, с. 306], [Гла, 8.4]

Определение 15.21. Пусть $\Sigma_3 = \{a,b,c\}$, $\vec{x} = (x_1,\ldots,x_n)$, $\vec{y} = (y_1,\ldots,y_n)$, где $x_i \in \{a,b\}^*$ и $y_i \in \{a,b\}^*$ для всех i. Обозначим через $\mathcal{K}_{\vec{x},\vec{y}}$ язык $\mathcal{L}_{\vec{x}} \cdot \{c\} \cdot \mathcal{L}_{\vec{y}}^{\mathrm{R}}$.

Лемма 15.22. *Язык $\mathcal{K}_{\vec{x},\vec{y}}$ является контекстно-свободным.*

Доказательство. Утверждение следует из теорем 9.13 и 9.11. □

Лемма 15.23. *Пусть $\Sigma_3 = \{a,b,c\}$, $\vec{x} = (x_1,\ldots,x_n)$, $\vec{y} = (y_1,\ldots,y_n)$, где $x_i \in \{a,b\}^*$, $y_i \in \{a,b\}^*$ и $x_iy_i \neq \varepsilon$ для всех i. Тогда язык $\mathcal{K}_{\vec{x},\vec{y}} \cap \{zcz^{\mathrm{R}} \mid z \in \Sigma_3^*\}$ является контекстно-свободным тогда и только тогда, когда постовская система соответствия $(\vec{x},\vec{y})$ не имеет решений.*

Доказательство. Пусть $(i_1,\ldots,i_k)$ — решение постовской системы соответствия $(\vec{x},\vec{y})$, где $x_iy_i \neq \varepsilon$ для всех i. Обозначим $u \rightleftharpoons ba^{i_k}ba^{i_{k-1}}\ldots ba^{i_1}$, $v \rightleftharpoons x_{i_1}\ldots x_{i_k}$ и $L_0 \rightleftharpoons \{u\}^* \cdot \{c\} \cdot \{v\}^* \cdot \{c\} \cdot \{v^{\mathrm{R}}\}^* \cdot \{c\} \cdot \{u^{\mathrm{R}}\}^*$. Легко проверить, что $u \neq \varepsilon$, $v \neq \varepsilon$ и язык L_0 является автоматным. Очевидно, $\mathcal{K}_{\vec{x},\vec{y}} \cap \{zcz^{\mathrm{R}} \mid z \in \Sigma_3^*\} \cap L_0 = \{u^mcv^mc(v^{\mathrm{R}})^mc(u^{\mathrm{R}})^m \mid m > 0\}$. Можно доказать (например, используя лемму 9.1), что язык $\{u^mcv^mc(v^{\mathrm{R}})^mc(u^{\mathrm{R}})^m \mid m > 0\}$ не является контекстно-свободным. Согласно теореме 9.16 язык $\mathcal{K}_{\vec{x},\vec{y}} \cap \{zcz^{\mathrm{R}} \mid z \in \Sigma_3^*\}$ также не является контекстно-свободным. □

Теорема 15.24. *Пусть $|\Sigma| \geqslant 2$. Тогда не существует алгоритма, позволяющего по произвольным контекстно-свободным грамматикам G_1 и G_2 над алфавитом Σ узнать, является ли контекстно-свободным язык $L(G_1) \cap L(G_2)$.*

Доказательство. Достаточно построить по постовской системе соответствия $(\vec{x},\vec{y})$, где $\vec{x} = (x_1,\ldots,x_n)$, $\vec{y} = (y_1,\ldots,y_n)$ и для всех i выполняется $x_i \in \{a,b\}^*$, $y_i \in \{a,b\}^*$ и $x_iy_i \neq \varepsilon$, контекстно-свободную грамматику G_1, порождающую язык $\mathcal{K}_{\vec{x},\vec{y}}$, и контекстно-свободную грамматику G_2, порождающую язык $\{zcz^{\mathrm{R}} \mid z \in \Sigma_3^*\}$. В силу леммы 15.23 неразрешимость рассматриваемой задачи сводится к неразрешимости проблемы соответствий Поста рассуждением, аналогичным приведённому в доказательстве теоремы 15.20. □

Лемма 15.25. *Рассмотрим алфавит* $\Sigma_3 = \{a, b, c\}$. *Язык* $\Sigma_3^* - (\mathcal{K}_{\vec{x},\vec{y}} \cap \{zcz^{\mathrm{R}} \mid z \in \Sigma_3^*\})$ *является контекстно-свободным.*

Доказательство. Обозначим $L_0 = \{w \in \Sigma_3^* \mid |w|_c = 1\}$. Язык $\Sigma_3^* - (\mathcal{K}_{\vec{x},\vec{y}} \cap \{zcz^{\mathrm{R}} \mid z \in \Sigma_3^*\})$ можно представить в виде объединения пяти контекстно-свободных языков

$$L_1 = \{w \in \Sigma_3^* \mid |w|_c \neq 3\},$$
$$L_2 = \{v_1cv_2cv_3cv_4 \mid v_1, v_2, v_3, v_4 \in \{a, b\}^*,\ v_1 \neq v_4^{\mathrm{R}}\},$$
$$L_3 = \{v_1cv_2cv_3cv_4 \mid v_1, v_2, v_3, v_4 \in \{a, b\}^*,\ v_2 \neq v_3^{\mathrm{R}}\},$$
$$L_4 = ((\Sigma_3^* - \mathcal{L}_{\vec{x}}) \cap L_0) \cdot \{c\} \cdot L_0,$$
$$L_5 = L_0 \cdot \{c\} \cdot ((\Sigma_3^* - \mathcal{L}_{\vec{y}})^{\mathrm{R}} \cap L_0).$$

□

Теорема 15.26. *Пусть* $|\Sigma| \geqslant 2$. *Тогда не существует алгоритма, позволяющего по произвольной контекстно-свободной грамматике* G *над алфавитом* Σ *узнать, является ли контекстно-свободным язык* $\Sigma^* - L(G)$.

Доказательство. Рассмотрим алфавит $\Sigma_3 = \{a, b, c\}$. Достаточно построить по постовской системе соответствия $(\vec{x}, \vec{y})$, где $\vec{x} = (x_1, \ldots, x_n)$, $\vec{y} = (y_1, \ldots, y_n)$ и для всех i выполняется $x_i \in \{a, b\}^*$, $y_i \in \{a, b\}^*$ и $x_iy_i \neq \varepsilon$, контекстно-свободную грамматику G, порождающую язык $\Sigma_3^* - (\mathcal{K}_{\vec{x},\vec{y}} \cap \{zcz^{\mathrm{R}} \mid z \in \Sigma_3^*\})$. В силу леммы 15.25 неразрешимость рассматриваемой задачи сводится к неразрешимости проблемы соответствий Поста рассуждением, аналогичным приведённому в доказательстве теоремы 15.20. □

Литература

[АхоСетУль] *Ахо А., Сети Р., Ульман Дж. Д.* Компиляторы: принципы, технологии и инструменты. М.: Вильямс, 2001. — 768 с.

[АхоУль] *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1: Синтаксический анализ. М.: Мир, 1978. — 612 с.

[Бра] *Братчиков И. Л.* Синтаксис языков программирования. М.: Мир, 1975. — 232 с.

[Гин] *Гинзбург С.* Математическая теория контекстно-свободных языков. М.: Мир, 1970. — 326 с.

[Гла] *Гладкий А. В.* Формальные грамматики и языки. М.: Наука, 1973. — 368 с.

[ГлаМел] *Гладкий А. В., Мельчук И. А.* Элементы математической лингвистики. М.: Наука, 1969. — 192 с.

[ГорМол] *Гордеев А. В., Молчанов А. Ю.* Системное программное обеспечение. СПб.: Питер, 2001. — 736 с.

[ГроЛан] *Гросс М., Лантен А.* Теория формальных грамматик. М.: Мир, 1971. — 294 с.

[КукБей] *Кук Д., Бейз Г.* Компьютерная математика. М.: Наука, 1990. — 384 с.

[Лал] *Лаллеман Ж.* Полугруппы и комбинаторные приложения. М.: Мир, 1985. — 440 с.

[ЛПИИ] Логический подход к искусственному интеллекту: От классической логики к логическому программированию. М.: Мир, 1990. — 432 с.

[Рей] *Рейуорд-Смит В. Дж.* Теория формальных языков. Вводный курс. М.: Радио и связь, 1988. — 128 с.

[Сал] *Саломаа А.* Жемчужины теории формальных языков. М.: Мир, 1986. — 159 с.

[СокКушБад] *Соколов В. А., Кушниренко О. Б., Бадин Н. М.* Формальные языки и грамматики. Задачи и упражнения. Ярославль: Ярославский государственный университет, 1993. — 55 с.

[ТраБар] *Трахтенброт Б. А., Барздинь Я. М.* Конечные автоматы (поведение и синтез). М.: Наука, 1970. — 400 с.

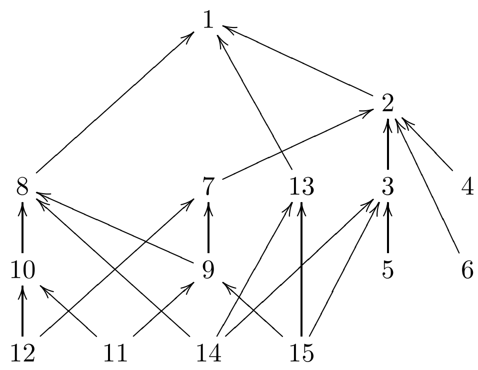
[ХопМотУль] *Хопкрофт Дж. Э., Мотвани Р., Ульман Дж. Д.* Введение в теорию автоматов, языков и вычислений, 2-е изд. М.: Вильямс, 2002. — 528 с.

[LewPap1] *Lewis H. R., Papadimitriou C. H.* Elements of the Theory of Computation. Prentice Hall, 1981. — 466 p.

[LewPap2] *Lewis H. R., Papadimitriou C. H.* Elements of the Theory of Computation. 2nd ed. Prentice Hall, 1998. — 361 p.

[Sip] *Sipser M.* Introduction to the Theory of Computation. PWS Publishing company, 1997. — 396 p.

Схема зависимости глав



Оглавление

Учебное издание

Пентус Анна Евгеньевна, Пентус Мати Рейнович

Теория формальных языков

М.: Издательство Центра прикладных исследований при механико-математическом факультете МГУ, 80 стр.

Оригинал-макет подготовлен авторами в пакете LaTeX.

Подписано в печать 08.12.2003.
Формат 60×90 1/16. Объём 5 усл. печ. л.
Заказ 5. Тираж 100 экз.

Изд-во ЦПИ при механико-математическом факультете МГУ.
119992, Москва, Ленинские горы.
Лицензия на издательскую деятельность ИД №04059 от 20.02.2001.

Отпечатано с оригинал-макета на типографском оборудовании механико-математического факультета и Франко-русского центра им. А. М. Ляпунова.