# 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 $\Sigma$ называется конечная последовательность элементов $\Sigma$.

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

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

**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 $\Sigma$, 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 $\Sigma$ обозначается $\Sigma^*$.

**Определение 1.12.** Множество всех непустых слов в алфавите $\Sigma$ обозначается $\Sigma^+$.

**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 $\Sigma$.

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

**Определение 1.23.** Пусть $L \subseteq \Sigma^*$. Тогда язык $\Sigma^* - L$ называется *дополнением* (complement) языка $L$ относительно алфавита $\Sigma$. 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 $L = \{a^k ba^l \mid 0 < k < l\}$, то $L^2 = \{a^k ba^l ba^m \mid 0 < k < l - 1,\ m > 1\}$.

**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 $w^{\mathrm{R}}$) 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 $\Sigma_1$ и $\Sigma_2$ — 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 $\Sigma$ — конечные алфавиты, $N \cap \Sigma = \varnothing$, $P \subset (N \cup \Sigma)^+ \times (N \cup \Sigma)^*$, $P$ конечно и $S \in N$. Здесь $\Sigma$ — *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 $\Sigma$ 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 $\Sigma$ — 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 $\alpha$, $\beta$, $\eta$, $\theta$ 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 $\omega_0$, $\omega_1$, ..., $\omega_n$ называется *выводом* (derivation) слова $\omega_n$ из слова $\omega_0$ 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 — $\Sigma$ 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 $\Sigma$.

**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 *$\varepsilon$-правилами*.

**Лемма 1.68.** *Каждая праволинейная грамматика является линейной. Каждая линейная грамматика является контекстно-свободной. Каждая контекстно-свободная грамматика без $\varepsilon$-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 $\Sigma^*$ 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 $\Sigma$ — конечный алфавит, $Q$ и $\Delta$ — 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 $\langle q_0, e_1, q_1, e_2, \dots, q_n\rangle$, 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*, $w_1 \dots w_n$ — *метка пути*.

**Замечание 2.4.** Для любого состояния $q \in Q$ существует путь $\langle q\rangle$. Его метка — $\varepsilon$, 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 $\varepsilon$.

**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 $L(M) = \{(ab)^n \mid n \geqslant 0\} \cup \{(ab)^n a \mid n \geqslant 0\}$.

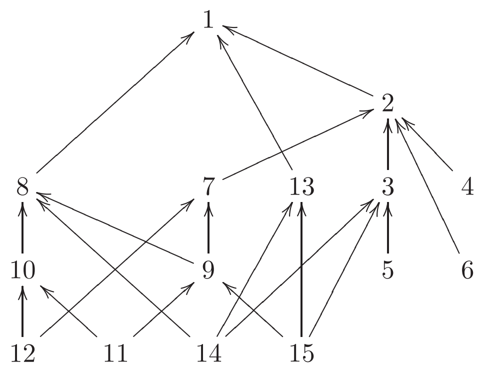
**Определение 2.11.** Два конечных автомата *эквивалентны*, если они распознают один и тот же язык.

**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.

## Схема зависимости глав




## Оглавление