

Конспект по дискретной математике  
II семестр

Коченюк Анатолий

31 мая 2021 г.



# Глава 1

## Дискретная теория вероятностей

### 1.1 Введение

**Определение 1** (Вероятностное пространство).

$\Omega$  – элементарные исходы, неделимые дальше.

$p$  – дискретная плотность вероятности.

$$p : \Omega \rightarrow [0, 1] \quad \sum_{\omega \in \Omega} p(\omega) = 1$$

**Замечание.** В случае дискретного вероятностного пространства  $|\Omega|$  – не более, чем счётное.

**Пример** (Честная монета).  $\Omega = \{0, 1\}$   $p(0) = p(1) = \frac{1}{2}$

**Пример** (Нечестная монета).  $\Omega = \{0, 1\}$   $p(1) = p, p(0) = q$  – различные числа.  $p + q = 1$

Ещё одно название – распределение Бернулли

**Пример** (Честная игральная кость).  $\Omega = \{1, 2, 3, 4, 5, 6\}$   $p(\omega) = \frac{1}{6}$

**Определение 2.** Событие, случайное событие –  $A \subseteq \Omega$

**Замечание.** Неправильное определение – то, что может произойти, а может не произойти.

---

$\emptyset \subseteq \Omega \quad \Omega \subseteq \Omega$  – примеры, когда никогда не происходит и всегда происходит

**Замечание.** Для не дискретного случая неверно, что любое подмножество  $\Omega$  это событие

**Определение 3.** Вероятность события  $P(A) = \sum_{\omega \in A} p(\omega)$   
 $p$  берёт элементарные исходы.  $P, \mathbb{P}$  – вероятность события

**Пример.** Событие  $E = \{2, 4, 6\}$   $P(E) = p(2) + p(4) + p(6) = \frac{3}{6} = \frac{1}{2}$   
 $O = \{1, 3, 5\}$

**Замечание.** Не существует вероятностного пространства с бесконечным числом равновероятных исходов

$$p(\omega) = 0 \quad \sum = 0$$

$$p(\omega) = a > 0 \quad \sum = a \cdot (+\infty) = +\infty$$

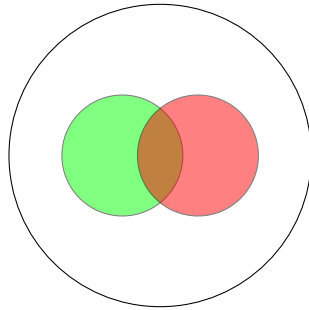
**Пример.** Событие  $B(IG) = \{4, 5, 6\}$   $P(B) = \frac{1}{2}$

**Определение 4** (Независимое событие). События  $A, B$  независимы, если  $P(A \cap B) = P(A) \cdot P(B)$

**Пример.**  $E \cap O = \emptyset \quad B \cap E = \{4, 6\}$

$$P(E \cap O) = 0 \quad P(O) \cdot P(B) = \frac{1}{4} \neq 0$$

$$P(B) \cdot P(E) = \frac{1}{4} \neq \frac{1}{3} = P(B \cap E)$$



$$P(A \cap B) = P(A) \cdot P(B)$$

$$\frac{P(A \cap B)}{P(B)} = \frac{P(A)}{P(\Omega)}$$

---

**Определение 5** (Условная вероятность).  $P(A|B) = \frac{P(A \cap B)}{P(B)}$

**Замечание.** Альтернативное определение независимости, не поддерживающее 0:  $P(A|B) = P(A)$

$$V = \{5, 6\}$$

$$P(V \cap E) = \frac{1}{6}$$

$$P(V) = \frac{1}{3} \quad P(E) = \frac{1}{2} \quad P(V) \cdot P(E) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6} = P(V \cap E)$$

**Определение 6** (Произведение вероятностных пространств).

$$\Omega_1, p_1 \quad \Omega_2, p_2$$

$$\Omega = \Omega_1 \times \Omega_2$$

$$p(\langle \omega_1, \omega_2 \rangle) = p_1(\omega_1) \cdot p_2(\omega_2)$$

**Теорема 1.**  $\forall A_1 \subseteq \Omega_1$  и  $\forall A_2 \subseteq \Omega_2$

$A_1 \times \Omega_2$  и  $\Omega_1 \times A_2$  – независимы

$$\begin{aligned} \text{Доказательство. } P(A_1 \times \Omega_2 \cap \Omega_1 \times A_2) &= P(A_1 \times A_2) = \sum_{\substack{a \in A_1 \\ b \in A_2}} p(\langle a, b \rangle) = \\ &= \sum_{a \in A_1} \sum_{b \in A_2} p_1(a) \cdot p_2(b) = \sum_{a \in A_1} p_1(a) \left( \sum_{b \in A_2} p_2(b) \right) = P_1(A_1) \cdot P_2(A_2) \quad \blacksquare \end{aligned}$$

**Определение 7.**  $A_1, A_2, \dots, A_n$

1. Попарно независимые  $A_i$  и  $A_j$  независимы

2. Независимы в совокупности  $\forall I \subseteq \{1, 2, \dots, n\} \quad P\left(\bigcap_{i \in I} A_i\right) = \prod_{i \in I} P(A_i)$

$$P(A_1 \cap A_2 \cap A_3) = P(A_1) \cdot P(A_2) \cdot P(A_3)$$

**Пример.** Кидаем две монеты  $\Omega = \{00, 01, 10, 11\}$

$A_1 = \{10, 11\}$   $A_2 = \{01, 11\}$   $A_3 = \{01, 10\}$  – независимы попарно, но не в совокупности

---

**Определение 8** (Формула полной вероятности).

$$\Omega = A_1 \cup A_2 \cup \dots \cup A_n \quad i \neq j \implies A_i \cap A_j = \emptyset$$

Совокупность таких  $A$ -шек называется полной системой событий.

Дано: вероятности  $P(A_i)$   $P(B|A_i)$  Найти:  $P(B)$

$$P(B) = \sum_{i=1}^n P(B \cap A_i) = \sum_{i=1}^n P(B|A_i) \cdot P(A_i)$$

– формула полной вероятности

Найти:  $P(A_j|B)$

$A_1$  – болен,  $A_2$  – здоров,  $B$  – положительный результат теста

$P(A_2|B)$

$$P(A_j|B) = \frac{P(A_j \cap B)}{P(B)} = \frac{P(B|A_j) \cdot P(A_j)}{\sum_{i=1}^n P(B|A_i) \cdot P(A_i)}$$

– формула Байеса

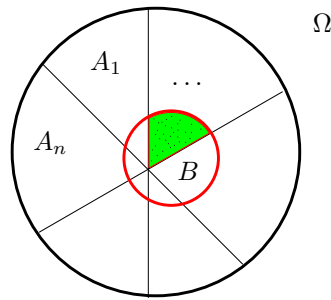


Рис. 1.1: В

## 1.2 Случайные величины

**Замечание.** Неправильное (наивное) определение – величина, принимающая случайное значение.

---

Она может быть константой. Что такое величина?

**Определение 9** (Случайная величина).  $\xi : \Omega \rightarrow \mathbb{R} - \mathbb{R}$ -значная функция

$\Omega, p$  – вероятностное пространство.

**Пример.** Если взять случайные текст длиной 1Кб. Вариантов текста очень много и бессмысленно их рассматривать отдельно, интересует какое-то свойство, величина.

Графы,  $2^{\binom{n}{2}}$  штук. Но нас интересует какая-то (численная) характеристика элементарного исхода.

**Пример.**  $D(ice) = \{1, 2, 3, 4, 5, 6\}$

$$\Omega = D^2 \quad p(\langle i, j \rangle) = \frac{1}{36}$$

$$\xi : \Omega \rightarrow \mathbb{R} \quad \xi(\langle i, j \rangle) = i + j$$

**Пример** (Случайные графы).  $G(4, \frac{1}{2})$  – случайный граф, 4 вершины, каждое ребро существует с вероятностью  $\frac{1}{2}$

$$\Omega = \mathbb{B}^6 \quad p(G) = \frac{1}{64}$$

$\xi(G)$  = количество компонент связности

**Пример.**  $\Omega = \{1, 2, 3, 4, 5, 6\}$   $\xi(w) = w$

**Пример.**  $\Omega = \{1, 2, 3, 4, 5, 6\}$   $E = \{2, 4, 6\}$

$$\chi_E(\omega) = \begin{cases} 1, & \omega \in E \\ 0, & \omega \notin E \end{cases} - \text{индикаторная случайная величина}$$

**Определение 10.**  $\Omega, p, \xi$

$$[\xi = i] = \{\omega | \xi(\omega) = i\} \subseteq \Omega$$

$$P([\xi = i]) = P(\xi = i) = f_\xi(i) \quad f : \mathbb{R} \rightarrow \mathbb{R}$$

$f_\xi(i) = P(\xi = i)$  – дискретная плотность вероятности случайной величины  $\xi$

$$F_\xi(i) : \mathbb{R} \rightarrow \mathbb{R} = P(\xi \leq i) - \text{функция распределения}$$

**Замечание.** Непрерывная vs Дискретная вероятность

**Пример.**

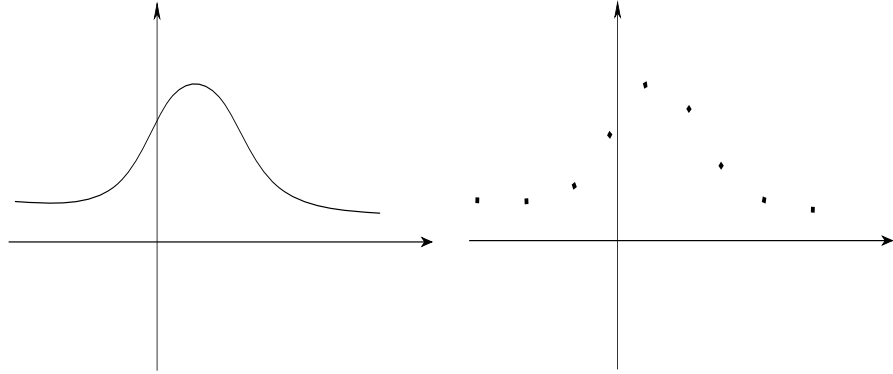


Рис. 1.2: непрерывная и дискретная вероятность

**Замечание.**  $\delta(x) = \begin{cases} 0, & x \neq 0 \\ +\infty, & x = 0 \end{cases}$

$$\int_{-\infty}^{+\infty} \delta(x) dx = 1$$

$$f_{\xi}(i) = P(\xi = i) \quad f_{\xi} = \int_{-\infty}^{+\infty} f_{xi}(x) = F_{\xi}(i)$$

$$f_{\xi}(x) = \sum_i P(\xi = i) f(x - i)$$

**Пример.**  $\Omega = \mathbb{B}^{1000} \quad p(\omega) = \frac{1}{2^{1000}}$

$\xi(w) = \text{число } 1 \text{ в } \omega$

$$|\text{множество значений } \xi| = 1001 \quad p(\xi = i) = \frac{\binom{1000}{i}}{2^{1000}}$$

**Замечание.** Случайные числа обозначаются строчными греческими или заглавными латинскими из конца алфавита (X, Z)

**Замечание** (Что можно делать со случайными величинами).  $\xi, \eta$  – функции

$\xi^2 \quad 2\xi \quad \xi + \eta \quad \xi \cdot \eta \quad \xi^\eta \quad \sin \xi \quad e^\eta \quad \frac{1+\xi}{\eta}$  (всё то же, что мы можем делать с функциями).

**Пример.**  $\Omega = D^2 \quad \xi_1(\langle i, j \rangle) = i \quad \xi_2(\langle i, j \rangle) = j$  – одинаково распределённые случайные величины



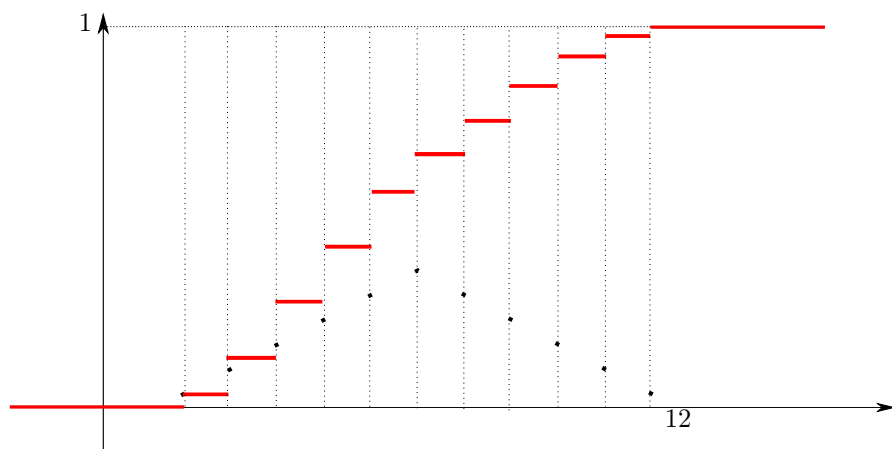


Рис. 1.3: Два-кубика (функция распределения)

**Пример.**  $\Omega = F$   $id(\omega) = \omega$

$1, 2, \dots, 6$  – каждый с вероятностью  $\frac{1}{6}$ . Другое вероятностное пространство относительно предыдущего примера, но всё равно одинаковое распределение.

$\xi = (i + j) = \xi_1 + \xi_2$   $\xi = (i + j) \% 6 + 1$  – у второй то же распределение, что у верхних, но она уже совсем другая.

**Определение 11.** Математическое ожидание

$$E_{\xi} = \sum_{\omega} p(\omega) \xi(\omega).$$

**Утверждение 1.**  $E_{\xi} = \sum_i i p(\xi = i)$

---

*Доказательство.*

$$\begin{aligned}
 E_{\xi} &= \sum_{\omega} p(\omega) \xi(\omega) \\
 &= \sum_i \sum_{\omega: \xi(\omega)=i} p(\omega) \cdot \xi(\omega) \\
 &= \sum_i \sum_{\omega: \xi(\omega)=i} p(\omega) \cdot i \\
 &= \sum_i i \sum_{\omega: \xi(\omega)=i} p(\omega) \\
 &= \sum_i iP(\xi = i)
 \end{aligned}$$

■

**Пример.**  $\Omega = D \quad \xi = id$

$$E_{\xi} = \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 + \frac{1}{6} \cdot 3 + \frac{1}{6} \cdot 4 + \frac{1}{6} \cdot 5 + \frac{1}{6} \cdot 6 = \frac{21}{6} = \frac{7}{2} = 3,5$$

**Пример.**  $\Omega = D^2 \quad \xi(\langle i, j \rangle) = i + j$

$$E_{\xi} = \frac{1}{36} \cdot (2 \cdot 1 + 3 \cdot 2 + 4 \cdot 3 + 5 \cdot 4 + 6 \cdot 5 + 7 \cdot 6 + 8 \cdot 5 + 9 \cdot 4 + 10 \cdot 3 + 11 \cdot 2 + 12 \cdot 1)$$

– здесь среднее значение оказалось наиболее частым, но так оно не всегда (пример с 3,5)

**Теорема 2.**  $E_{\lambda\xi} = \lambda E_{\xi}$

$$E(\xi + \eta) = E_{\xi} + E_{\eta}$$

*Доказательство.*  $E_{\lambda\xi} = \sum_{\omega} p(\omega) \lambda\xi(\omega) = \lambda E_{\xi}$

$$E(\xi + \eta) = \sum_{\omega} p(\omega) (\xi(\omega) + \eta(\omega)) = E_{\xi} + E_{\eta}$$

■

**Утверждение 2.** Если  $\xi$  и  $\eta$  одинаково распределены, то  $E_{\xi} = E_{\eta}$

**Пример.** Бросим кубик один раз,  $\xi_1$  – что выпало сверху,  $\xi_2$  – что выпало снизу

$E(\xi_1 + \xi_2) = 7$ . – не играет роли как числа друг относительно друга расположены.

---

# МАТОЖИДАНИЕ ЛИНЕЙНО ВСЕГДА

**Пример.**  $\Omega = S_n$   $p(\omega) = \frac{1}{n!}$

$\xi(\pi) = |\{i | \pi[i] = i\}|$   $0 \dots n$ , кроме  $n-1$

$$E_\xi = \sum_{j=1}^n \xi_j = 1$$

$$\xi_i(\pi) = \begin{cases} 1, & \pi[i] = i \\ 0, & \text{иначе} \end{cases}$$

$$E_{\xi_i} = \frac{1}{n}$$

$$\xi = \sum_{j=1}^n \xi_j$$

## 1.3 Независимые случайные величины

**Определение 12** (удобное). Случайные величины  $\xi$  и  $\eta$  независимы, если события  $[\xi = \alpha]$  и  $[\eta = \beta]$  – независимы  $\forall \alpha, \beta$

**Определение 13** (нормальное).  $[\xi \leq \alpha]$  и  $[\eta \leq \beta]$  – независимы для  $\forall \alpha, \beta$

**Пример.**  $\Omega = \Omega_1 \times \Omega_2$

$$\xi_1(\langle \omega_1, \omega_2 \rangle) = f(\omega_1)$$

$$\xi_2(\langle \omega_1, \omega_2 \rangle) = g(\omega_2)$$

$A$  и  $B$  независимы,  $\chi_A, \chi_B$  – независимы

**Теорема 3.**  $\xi, \eta$  – независимы  $\implies E(\xi \cdot \eta) = E_\xi \cdot E_\eta$

*Доказательство.*  $E\xi \cdot \eta = \sum_{\alpha} \alpha \cdot P(\xi \cdot \eta = \alpha) = \sum_{i,j} \alpha P([\xi = i] \cap [\eta = j]) =$   
 $\sum_i \sum_j i j P(\xi = i) P(\eta = j) = E_\xi E_\eta$

$$i \cdot j = \alpha \quad i \in R_\xi \quad j \in R_\eta$$

■

**Пример.**  $\Omega = \{0, 1\}$   $p = \frac{1}{2}$   $\xi(i) = 2i$   $E_\xi = 1$

$\Omega = S_n$   $p = \frac{1}{n!}$   $\xi$  = число неподвижных точек  $E_\xi = 1$

Матожидание одно, но ведут себя совершенно по разному.

**Определение 14** (Дисперсия).  $D_\xi = Var(\xi)$

$$D_{xi} = E(\xi - E_\xi)^2 = E(\xi^2 - 2\xi E_\xi + (E_\xi)^2) = E_{xi}^2 - 2E_\xi E_\xi + (E_\xi)^2 = E_{\xi^2} - (E_\xi)^2$$

**Теорема 4.**  $D_{c\xi} = c^2 D_\xi$

Если  $\xi$  и  $\eta$  независимы, то  $D_{\xi+\eta} = D_\xi + D_\eta$

*Доказательство.* Упражнение ■

Вспомним.  $\xi, \eta : \Omega \rightarrow \mathbb{R}$

$$F_\xi(a) = P(\xi \leq a)$$

$$f_\xi(a) = P(\xi = a)$$

$$F_\xi(a) = \sum_{b \leq a} f_\xi(b)$$

$$E_\xi = \sum_{\omega \in \Omega} p(\omega) \xi(\omega) = \sum_a a \cdot P(\xi = a)$$

$$E(\xi + \eta) = E_\xi + E_\eta \quad \blacksquare$$

$E(\xi - E_\xi) = E_\xi - EE_\xi = 0$  матожидание отклонения от матожидания равно нулю..

Хочется смотреть насколько величина отклоняется от своего матожидания. Для этого используется понятие дисперсии:

$$D_\xi = E(\xi - E_\xi)^2 = E\xi^2 - (E\xi)^2$$

$$D(\xi + \eta) = E\xi^2 + E\eta^2 + 2E\xi\eta - (E\xi)^2 - (E\eta)^2 - 2E\xi E\eta = D_\xi + D_\eta + 2(E_{\xi\eta} - E_\xi E_\eta).$$

В случае независимых случайных величин дисперсия линейна. Иначе она отличается на ковариацию:

**Определение 15** (Ковариация).

$$Cov(\xi, \eta) = E_{\xi\eta} - E_\xi E_\eta.$$

$$D_\xi = Cov(\xi, \xi)$$

---

**Определение 16** (Корреляция).

$$Corr(\xi, \eta) = \frac{Cov(\xi, \eta)}{\sqrt{D_\xi D_\eta}}.$$

**Теорема 5.** Корреляция двух случайных величин лежит между -1 и 1.

$$-1 \leq Corr(\xi, \eta) \leq 1.$$

*Доказательство.*  $\alpha = \xi - \lambda\eta$

$$D_\alpha = E_{\xi^2} - 2\lambda E_{\xi\eta} + \lambda^2 E(\eta^2) - (E(\xi))^2 + 2\lambda E_\xi E_\eta - \lambda^2 (E_{\eta^2}) \geq 0$$

$$D_\xi + 2\lambda Cov(\eta, \eta) + \lambda^2 D_\eta$$

$$4Cov(\xi, \eta)^2 - 4D_\xi D_\eta \leq 0 \quad \blacksquare$$

## 1.4 Хвостовые неравенства

$$\xi \quad E\xi = 10 \quad \xi \geq 0$$

$$P(\xi \geq 100) < \frac{1}{10}$$

**Теорема 6** (Неравенство Маркова).  $\xi \neq 0 \quad \xi \geq 0 \quad P(\xi \geq a \cdot E_\xi) \leq \frac{1}{a}$

*Доказательство.*  $E_\xi = \sum_v v \cdot P(\xi = v) = \sum_{v < a \cdot E_\xi} v \cdot P(\xi = v) + \sum_{v \geq a \cdot E_\xi} v \cdot P(\xi = v) \geq 0 + a \cdot E_\xi \cdot \sum_{v \geq a \cdot E_\xi} P(\xi = v) = a \cdot E_\xi \cdot P(\xi \geq a \cdot E_\xi) \quad \blacksquare$

**Пример.**  $a = \frac{c}{E_\xi} \quad P(\xi \geq c) \leq \frac{E_\xi}{c}$

$$D_\xi = E(\xi - E\xi)^2$$

$$\eta = (\xi - E_\xi)^2$$

$$P((\xi - E_{xi})^2 \geq a^2 \cdot D_\xi) \leq \frac{1}{a^2}$$

$\sigma = \sqrt{D_\xi}$  – среднее квадратичное отклонение

**Теорема 7** (Неравенство Чебышева).  $P(|\xi - E_\xi| \geq a\sigma) \leq \frac{1}{a^2}$

$$P(|\xi - E_\xi| \geq c) \leq \frac{D_\xi}{c^2}$$

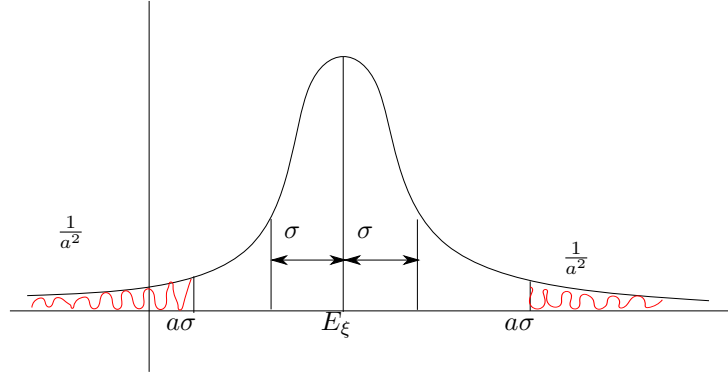


Рис. 1.4: drawing

**Задача 1** (10 монет, найти количество “1”).

$$E\xi = 5 \quad D\xi = 2,5$$

$$P(\xi \leq 0) \leq P(|\xi - E\xi| \geq 5) \leq \frac{2,5}{25} = \frac{1}{10}$$

**Замечание.** С одной стороны, у неравенства есть плюс: оно **всегда** работает; всегда (!)

С другой — иногда оценки получаются, очень грубыми. В нашем примере ответ  $\leq \frac{1}{10}$ , а в жизни —  $\frac{1}{1024}$ .

**Пример.** Нечестная монета  $p \neq \frac{1}{2}$ . Хотим выяснить чем чаще выпадает.

Бросили:  $c$  единиц,  $n - c$  нулей. Предположим, что  $c < \frac{n}{2}$   $p > \frac{1}{2}$   $pn > \frac{n}{2}$

$$P(\xi = c) \leq P(\xi \leq c) \leq P(|\xi - pn| \leq pn - c) \leq P(|\xi - pn| \leq \frac{n}{2} - c) \leq \frac{n}{4 \cdot (\frac{n}{2} - c)^2}$$

**Теорема 8** (Граница Чернова (без доказательства)).  $\xi_i$   $P(\xi_i = 1)$   $P(\xi_i = 0) = 1$

$$\xi = \sum_{i=1}^n \xi_i \quad E\xi = np = \mu$$

$$P(|\xi - \mu| \geq \delta\mu) < e^{-\mu \frac{\delta^2}{3}}$$

**Пример.** Случайная величина  $\xi$ . Хотим узнать матожидание. Проведём эксперимент  $n$  раз:  $\xi_1, \xi_2, \dots, \xi_n$

$$P\left(\left|\frac{\sum \xi_i}{n} - E_\xi\right| > c\right) \leq \frac{D_\xi}{n\varepsilon^2}.$$

$$\xi : \Omega \rightarrow \mathbb{Z}^+$$

$$E_\xi = \sum_{i=0}^n i \cdot P(\xi = i) = \sum_{i=0}^n (P(\xi \geq i) - P(\xi \geq i+1)) = \sum_{i=1}^n P(\xi \geq i)$$

## 1.5 Теория информации

**Определение 17** (Что такое информации).  
Информация = – неопределённость

неопределённость Н1. Что-то узнали, стала неопределённость Н2. полученная информация  $I = \text{Н1} - \text{Н2} = -\Delta \text{Н}$

Хочется убрать наблюдателя, нас, из определения, чтобы не было кого-то, кто узнаёт и меняет неопределённость. Надо ввести объективную модель:

**Определение 18** (Случайный источник).  $\Omega$  – вероятностное пространство.  
Есть исходы  $p_1, p_2, \dots, p_n$

Чёрный ящик с красной кнопкой и дисплеем. Основан на вероятностном пространстве

$$\xi_1, \xi_2, \dots, \xi_m \dots$$

$$P(\xi_i = a) = p_a \quad a = 1 \dots n$$

Случайный источник  $p_1, p_1, \dots, p_n$ . Хотим померять сколько информации содержится в одном результате эксперимента.

$$H(p_1, p_2, \dots, p_n) : RS(randomsources) \rightarrow \mathbb{R}^+$$

Частный случай  $p_i = \frac{1}{n}$

$$h(n) = H\left(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n}\right)$$

$$1. \quad h(n+1) > h(n)$$

2.

**Пример.**  $\Omega = \{(1, 1), (1, 2), \dots, (1, m_1)(2, 1), \dots, (2, m_2), \dots, (k, 1), \dots, (k, m_k)\}$

$$n = m_1 + m_2 + \dots m_n$$

$$p(i, j) = 1_{ij} \quad p_i = \sum_{j=1}^{m_k} q_{ij}$$

Первый ряд  $(1, *) - p_1$ . Второй  $p_2 \dots$  Последний  $p_k$

Если случайный источник показывает только первое число это эквивалентно  $H(p_1, p_2, \dots, p_n)$

Теперь представим, что мы сначала узнаём первую компоненту, а потом открываем вторую

$$\sum_{i=1}^k p_i H\left(\frac{q_{i1}}{p_i}, \dots, \frac{q_{im_i}}{p_i}\right)$$

Если провести эксперимент сразу, получим  $q_{11}, \dots, q_{im_i}$

$$q_{ij} = p_i q_{ij}$$

$$H(p_1 r_{11}, p_1 r_{12}, \dots, p_1 r_{1k_1}, p_2 r_{21}, \dots, p_k r_{km_k}) = H(p_1, p_2, \dots, p_k) + \sum_{i=1}^k p_i H(r_{i1}, \dots, r_{im_i}).$$

3. Для фиксированного  $n$   $H$  непрерывная как функция  $\mathbb{R}^n \rightarrow \mathbb{R}$

$$\textbf{Теорема 9. } H(p_1, p_2, \dots, p_n) = - \sum_{i=1}^n p_i \log p_i$$

**Лемма 1.**  $h(nm) = H(n) + h(m)$  Следует из второго свойства

$$\textit{Доказательство. } k = n \quad m_i = m \quad p_i = \frac{1}{n} \quad q_{ij} = \frac{1}{nm} \quad r_{ij} = \frac{1}{m}$$

$$h(nm) = H(q_{11}, q_{12}, \dots, q_{nm}) = H\left(\frac{1}{n}, \dots, \frac{1}{n}\right) + \sum_{i=1}^n \frac{1}{n} H\left(\frac{1}{m}, \dots, \frac{1}{m}\right) = h(n) + h(m) \quad \blacksquare$$

**Определение 19.**  $h(2) = \alpha$  (может с точностью до мультипликативной константы задать)

$$\textbf{Лемма 2. } h(2^k) = k\alpha$$

$$\textbf{Лемма 3. } h(n) = \alpha \log_2 n$$

$$\textit{Доказательство. } 2^i \leq n^r < 2^{i+1} \quad r \in \mathbb{N}$$

$$h(i) \leq h(n^r) < \alpha(i+1)$$



$$\alpha \cdot i \leq r \cdot h(n) < \alpha(i+1)$$

$$\alpha \cdot \frac{i}{r} \leq h(n), \alpha \frac{i+1}{r}$$

$$i \leq r \log_2 n < i+1$$

$$\alpha \frac{i}{r} \leq \alpha \log_2 n < \alpha \frac{i+1}{r}$$

$$\forall r \quad |h(n) - \alpha \log_2 n| \leq \frac{\alpha}{r}$$

$$\implies h(n) = \alpha \log_2 n$$

■

*Доказательство теоремы.* Рациональный  $p_i = \frac{a_i}{b}$   $m_i = a_i$   $r_{ij} = \frac{1}{a_i}$   $q_{ij} = \frac{1}{b}$   $q_{ij} = p_i r_{ij}$

$$H\left(\underbrace{\frac{1}{b}, \dots, \frac{1}{b}}_b\right) = H(p_1, \dots, p_k) + \sum_{i=1}^k p_i H\left(\underbrace{\frac{1}{a_i}, \dots, \frac{1}{a_i}}_{a_i}\right)$$

$$\left(\sum_{i=1}^k p_i\right) h(b) = H(p_1, \dots, p_k) + \sum_{i=1}^k p_i h(a_i)$$

$$H(p_1, \dots, p_k) = \sum_{i=1}^k p_i (\alpha \log_2 b - \alpha \log_2 a_i)$$

Функция непрерывна и она верна для рациональных, следовательно она верна для всех

■

**Замечание.**  $h(2) = \alpha$  – бит

А теперь мы хотим перевести определение информации на неслучайный источник

*Ответ 1.* Это тогда будет не совсем корректно с математической точки зрения. Когда смотришь на конкретные детерминированные данные.

■

*Ответ 2.* Изучение среднего не совсем антинаучное занятие. Внешне оно ведёт себя как случайные величины.

■

**Пример.** Есть строка  $s$ , в которой мы хотим померить информацию.

$$s \in \Sigma^* \quad n = |\Sigma|$$

$$|s| = l \quad f_i - \text{количество символов } c_i \text{ в строке } s$$

$$p_i = \frac{f_i}{L}$$

Допустим, что символы выдаёт случайный источник, который выдал символы  $s_1, s_2, \dots, s_L$ . Статистически эта строка похожа на  $s$ . \*натягивание на глобус\* Допустим, что количество информации в строке  $s$  равно количеству в строке  $\tilde{s}$

$$I(\tilde{s}) = J \cdot H(p_1, p_2, \dots, p_n) = -L \sum_{i=1}^n p_i \log_2 p_i$$

Вспомним арифметическое кодирование

$q = A(s)$  – длина арифметического кодирования

$$\begin{aligned} A(s) &\leq -\log_2(b_L - a_L) = -\log_2(p_{s_1} \cdot \dots \cdot p_{s_L}) = -\log_2\left(\prod_{i=1}^n p_i^{f_i}\right) = -\sum_{i=1}^n \underbrace{\frac{L}{p_i}}_{p_i} \log_2 p_i = \\ &= I(\tilde{s}) = L \cdot H(p_1, p_2, \dots, p_n) \end{aligned}$$

**Теорема 10.** Длина кода, после арифметического кодирования не превышает энтропию Шеннона

**Замечание.** Арифметическое кодирование асимптотически оптимально среди тех, которые не учитывают взаимное расположение символов.

**Пример** (Нижняя оценка для сортировки). Пусть  $a_1, \dots, a_n$  – перестановка и мы хотим её отсортировать

**Утверждение 3.** От одного сравнения мы получаем не больше 1 бита информации

Рассмотрим все перестановки. В каждой содержится

$$\log_2 n! = \sum_{i=1}^n \log_2 i \geq \sum_{i=\frac{n}{2}}^n \log_2 \frac{n}{2} = \Omega(n \log n)$$

## 1.6 Цепи Маркова

$b = (b_1, b_2, \dots, b_n)$  –  $b_i$  вероятность находиться в состоянии  $i$

$C \quad c = (c_1, c_2, \dots, c_n)$  – случайная величина после одного перехода

Матрица перехода  $p_{ij}$  – вероятность перейти из  $i$  в  $j$

$$P = \begin{bmatrix} 0 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$c_i = P(C = i) = \sum_{j=1}^n P(x = i | B = j) P(B = j) = \sum_{j=1}^n p_{ji} \cdot b_j$$

$b^0 = (1, 0, 0, 0)$  – нулевой шаг

$$b^1 = (0, \frac{1}{2}, \frac{1}{2}, 0)$$

---

Рассмотрим судьбу м.ц. после поглощения. Жизнь происходит внутри одной сильно связанной компоненты – эргодического класса.

1.  $d > 1$  длина любого цикла кратна  $d$ . Циклический класс
2.  $\text{НОД}(\text{длин всех циклов}) = 1$ .

**Теорема 11** (Эргодическая для регулярных цепей). М.ц. такова, что  $p_{ij} > 0 \forall i, j$

Тогда  $\exists b \quad \forall b^0 \quad b^0 P^n \rightarrow b$

( $b$  удовлетворяет равенству  $b = bP$ )

*Доказательство.*  $(b^0 A)_i = \sum_{j=1}^n b_j^0 \cdot A_{ji} = \left( \sum_{j=1}^n b_j^0 \right) \tilde{a}_i = \tilde{a}_i$

$$\square \forall j \quad a_{ji} = \tilde{a}_i$$

$P^n \rightarrow A$ , которая удовлетворяет условию выше.

$$m_i^t = \min_j (P^t)_{ji} \quad M_i^t = \min_j (P^t)_{ji}$$

$$M_i^t - m_i^t \rightarrow 0$$

$$\delta = \min_{i,j} \delta > 0$$

$$\begin{aligned} P_{ji}^{t+1} &= \sum_{k=1}^n P_{jk}^t P_{ki} \\ &\leq \overbrace{\sum_{k=1}^n P_{jk}^t M_i^t}^1 + P_{j \text{ posMin}} (m_i^t - M_i^t) \\ &\leq M_i^t + \delta (m_i^t - M_i^t). \end{aligned}$$

Аналогично с максимумом, оцениваем всё снизу минимумов, кроме максимума

$$M_i^{t+1} \leq M_i^t + \delta (m_i^t - M_i^t).$$

$$-m_i^{t+1} \leq -m_i^t + \delta (m_i^t - M_i^t).$$

$$M_i^{t+1} - m_i^{t+1} \leq (M_i^t - m_i^t) (1 - 2\delta) \leq (1 - 2\delta)^{t+1} \rightarrow 0.$$

---

Теперь у  $b = bP$

$$(I - P)b = 0$$

$$\text{Rg}(I - P) = n - 1$$

$$\sum b_i = 1$$

$$P^{2^c}$$

$$bP^n 0 > b \quad bP^{n+1} \rightarrow bP$$

$$b = bP$$



Вернёмся к вопросу что происходит после поглощения.

$$\triangleleft \text{эргодический класс } A \quad \tilde{p} = \sum_{a \in A} (b^0 N R)_a$$

$$\tilde{b}^0 = (b^0 N R)_{A - \frac{1}{p}}$$

$$\exists \text{ предельное } b : \quad \tilde{b}^0 A^n \rightarrow b$$

Конечное распределение  $b\tilde{p}$

Скрытые Марковские модели. Мы решали до этого прямую задачу – брали м.ц. с известными матрицами перехода и смотрели на их характеристики.

Есть обратная: Есть состояние и мы хотим узнать матрицу перехода.

Ещё задача: Есть немарковский процесс и мы хотим аппроксимировать его марковским.

## 1.7 Формальные языки

Алфавит –  $\Sigma$ , конечное непустое множество

$$\text{слово, цепочка, строка } \Sigma^* = \bigcup_{k=0}^{\infty} \Sigma^k$$

Формальный язык  $L \subseteq \Sigma^*$

<TODO>

**Определение 20.** Описание языка – слово конечной длины

Всего “описаний” счётное множество

- Распознавание – по слову возвращаем булевский флаг – есть слово в нашем языке или нет
- Порождение – описывает как породить возможно бесконечное количество слов

**Определение 21.** Перечисление слов.  $\{01, 011, 10, 1010\}$ . Но так можно описать только конечные языки (содержащие конечное количество слов)

**Пример.** Правильные скобочные последовательности  $\varepsilon$  – псп

$A, B$  – псп  $\implies AB$  – псп

$A$  – псп  $\implies (A)$  – псп

Это порождение. Можно описать распознаванием: баланс в любой момент неотрицательный, баланс в конце = 0

Между этими способами есть логический переход:

- Распознавать  $\rightarrow$  Порождать. порождаем всё, что можем распознать
- Обратно: распознаём всё, что в какой-то момент порождаем

**Пример.** C++ без ограничений по памяти. Пограмма  $P$

$L = \{\omega \mid p(\omega) = 1\}$

### 1.7.1 Регулярные = Автоматные языки

**Определение 22.** Конкатенация:  $\alpha \in \Sigma^k, \beta \in \Sigma^l \implies \alpha\beta \in \Sigma^{k+l}$

$$\gamma = \alpha\beta \quad \gamma_i = \begin{cases} i \leq k & \implies \alpha_i \\ i > k & \implies \beta_{i-k} \end{cases}$$

$$\alpha(\beta\gamma) = (\alpha\beta)\gamma \quad \alpha\varepsilon = \varepsilon\alpha = \alpha$$

**Пример.**  $AB = \{x \mid x = yz, y \in A, z \in B\}$

$A = \{0, 01\} \quad B = \{0, 10\}$

$AB = \{00, 010, 0110\}$

Базовые операции:

1. Объединение  $A \cup B$
2. Конкатенация  $AB$

Возведение в степень  $A^k = \underbrace{AA \dots A}_k \quad A^0 = \{\varepsilon\}$

3. Замыкание Клини  $A^* = \bigcup_{k=0}^{\infty} A^k$

---

**Определение 23.**  $Reg_0 = \{\emptyset, \{\varepsilon\}, \{c\} \mid c \in C\}$

$$Reg_{i+1} = Reg_i \cup \{A \cup B, AB, A^* \mid A, B \in Reg_i\}$$

$$Reg_1 = \{\emptyset, \varepsilon, a, b, \dots, \{a, b\}, \{a, \varepsilon\}, \dots, ab, aa, \dots, \{\varepsilon, a, aa, aaa, \dots\}, \dots, \{\varepsilon b, bb, bbb, \dots\}\}$$

$$Reg = \bigcup_{k=0}^{\infty} Reg_k$$

**Лемма 4.**  $A, B \in Reg$ :

1.  $A \cup B \in Reg$
2.  $AB \in Reg$
3.  $A^* \in Reg$

$$A \in Reg_i, B \in Reg_j$$

Они все принадлежат  $Reg_{\max\{i,j\}+1}$

**Определение 24.** Назовём семейство языков  $X \in Good$   $X \subseteq 2^{\Sigma^*}$

$$X = \text{set} \langle lang \rangle$$

Good:  $\text{set} \langle \text{set} \langle lang \rangle \rangle$

1.  $Reg_0 \in X$
2.  $X$  замкнуто относительно  $A \cup B, AB, A^*$ , а.и.

$$A, B \in X \implies AB, A \cup B, A^* \in X \quad A, B : lang.$$

**Теорема 12.**  $Reg = \bigcap_{u \in Good} U$

*Доказательство.* to be written



---

**Определение 25** (Описание). •  $\emptyset$   $\varepsilon$   $c$

•  $A \alpha B \beta$ :

- $AB \alpha \beta$  – средний приоритет
- $A \cup B \alpha | \beta$  – минимальный приоритет
- $A^* \alpha^*$  – максимальный приоритет

$$\Sigma = \{0, 1\}$$

$(0 | 11)^*$  – язык в котором единицы идут парами

Такие описания называют академическими регулярными выражениями

$$\alpha^+ = \alpha \alpha^*$$

$$\alpha^k = \underbrace{\alpha \alpha \dots \alpha}_k$$

**Пример.**  $0^* | (0^* 10^* 10^*)^*$  – язык, содержащий чётное число единиц

**Пример.** Проверка чётное ли число единиц

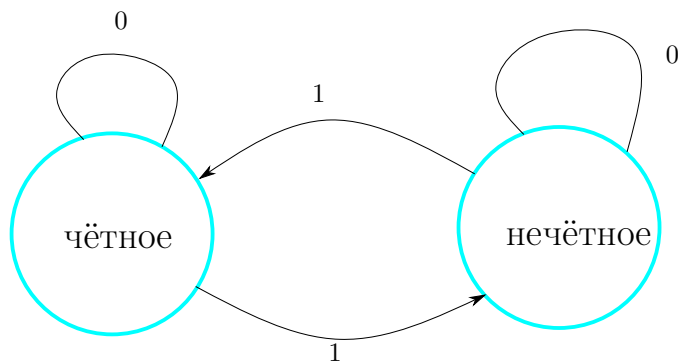


Рис. 1.5: check-one

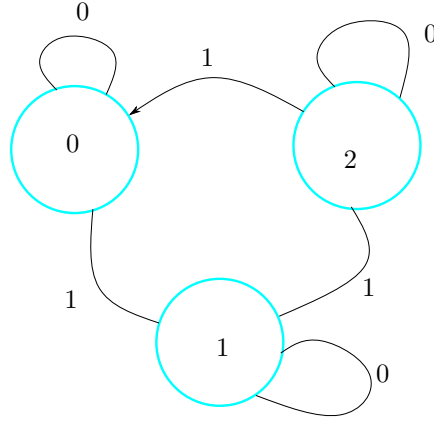


Рис. 1.6: check-div3

**Определение 26.** Детерминированный Конечный Автомат ДКА DFA

$$A = \langle \Sigma, Q, S \in \Sigma, T \subseteq Q, \delta : Q \times \Sigma \rightarrow Q \rangle.$$

- $\Sigma$  – алфавит
- $Q$  – конечное множество состояний
- $S$  – начальное состояние
- $T$  – допускающие состояния
- $\delta$  – функция переходов

$$Snap = Q \times \Sigma^*$$

Пееход:

1.  $\alpha = c\beta \quad c \in \Sigma$
2.  $r = \delta(q, c)$

**Пример.**  $\langle e, 0101 \rangle \vdash \langle e, 101 \rangle \vdash \langle o, 01 \rangle \vdash \langle o, 1 \rangle \vdash \langle e, \varepsilon \rangle$

$$\mathcal{L}(A) = \{\omega \mid \langle s, \omega \rangle \vdash^* \langle t, \varepsilon \rangle, t \in T\}$$

**Теорема 13** (Клини).  $Reg = Aut$

$$Aut = \{X \mid \exists \text{ ДКА } A : X = L(A)\}$$



---

## 1.8 Недетерминированный конечный автомат

$x$  – допускается Недетерминированным Конечным Автоматом  $\iff \exists$  последовательность переходов по символам  $x$ , заканчивающаяся в допускающем состоянии

$L$  – формальный язык.  $L \subseteq \Sigma^*$

Артур:  $x \mapsto x \in L$ ?

Мерлин: Убедить Артура, что  $x \in L$

**Замечание.** Артуру в случае неопределённости выгодно слушать Мерлина.

Если  $x \notin L$ , то Мерлин не сможет испортить своими советами, потому что в автомате просто нет такой последовательности, на которую можно направить, чтобы попасть в допускающее

Если  $x \in L$ , то, внезапно, интересы Артура и Мерлина совпадают

**Замечание** (интерпретация через миры). На каждом шаге, где недетерминирован следующий шаг, создаётся два мира, на каждый из шагов. Если хотя бы в одном дошли до допускающего, то слово принадлежит.

**Определение 27** (НКА).  $(\Sigma, Q, S \subseteq Q, T \subseteq Q, \delta : Q \times E \rightarrow 2^Q)$

Стартовых состояний может быть несколько, хотя почти никогда не нужно

Состояние –  $\langle q, x \rangle \quad q \in Q, \quad x \in X^*$

$\langle q, x \rangle \vdash \langle r, y \rangle$

1.  $x = cy, \quad c \in \Sigma$

2.  $r \in \delta(q, c)$

$x$  – допускается  $A$ , если  $\langle s, x \rangle \vdash^* \langle t, \varepsilon \rangle, \quad t \in T$

**Пример.**

```
class DFA {
    // 0 .. n-1 -- Q; 0 .. c-1 -- Sigma
    s : int
    t : vector<bool>(n)
    delta: vector<vector<int>>> (n,c)

    bool accept(x) {
        cur = s
        for (i = 0 .. len(x) - 1)
```

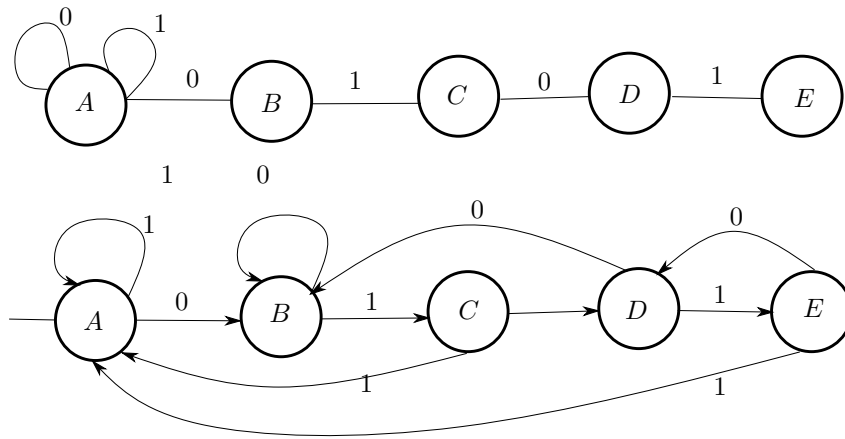


Рис. 1.7: auto

```

        cur = delta[cur][x[i]
    return t[cur]
}
}
O(len(x))

```

## 1.9 Динамическое программирование

```

class NFA {
    // 0 ... n-1 -- Q; 0 .. c-1 -- Sigma
    s : int
    t : vector<bool>(n)
    delta : vector<vector<set<int>>>(int)

    can[i][q] -- можно ли прочитав i символов x'а оказаться в состоянии q

    bool accept(x):
        can[0][s] = true
        for (i = 0 .. len(x) - 1)
            for (q = 0 .. n-1)
                if (can[i][q])
                    for (r : delta[q][x[i])
                        can[i+1][r] = true
        for (q = 0 .. n-1)
            if (can[len(x)][q] && t[q])
                return true

```

---

}  
 $O(\text{len}(x) \cdot (n^2 + m))$

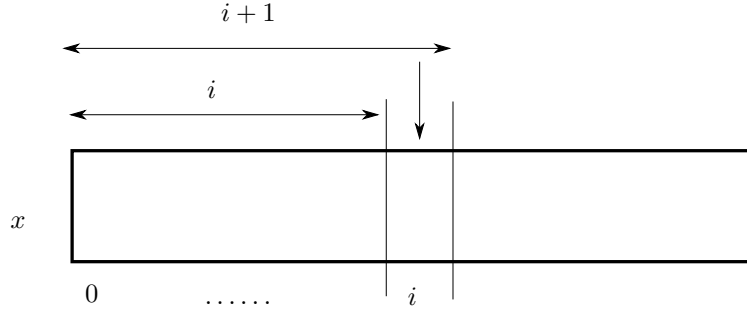


Рис. 1.8: nfa-step

**Утверждение 4.** Для  $L \exists$  НКА  $A_n \iff$  для  $L \exists$  ДКА  $A_D$

*Доказательство.*  $\forall$  ДКА является частным случаем НКА

$\Leftarrow$  очевидно

$\Rightarrow$  Алгоритм Томпсона

```
next(a : vector<bool>, c) vector<bool>
    res = vector<bool>(n)
    for (q = 0 .. n-1)
        if a[q]
            for r : delta[q][c]
                res[r] = true
    return res
```

```
bool accept(x)
    can[0][s] = true
    for (i = 0 .. len(x) - 1)
        can[i+1] = next(can[i], x[i])
```

$(\Sigma, Q_D = 2^{Q_N}, \{s\}, T_D = \{A \mid A \cap T_n \neq \emptyset\}, \delta_D(A, c) = \{r \mid \exists q \in A, r \in \delta_N(q, c)\})$

■

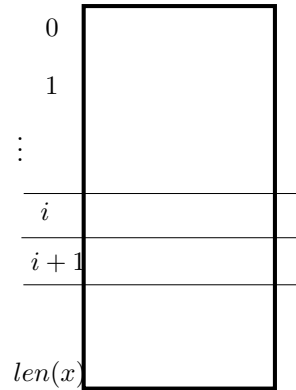


Рис. 1.9: tomps

**Замечание.** Казалось бы, вот Детерминированный автоматы такие хорошие, зачем нужны другие? Но он имеет экспоненциальное количество состояний (верхняя оценка) по сравнению с Недетерминированным

Но в реальности можно улучшать, убирая состояния, которые недостижимы (например  $\{B, C\}$  может не встречаться одновременно никогда) (в недетерминированном)

Иначе можно начать со стартовых состояний и делать очередь всех состояний, в которых мы можем быть. Именно такую конструкцию обычно и называют Алгоритмом Томпсона.

Конструкция описанная выше называется Конструкцией подмножеств.

## 1.10 $\varepsilon$ -НКА

Разрешим на переходе писать не символ, а  $\varepsilon$ . Переходя по нему строка на входе не меняется

**Пример.**  $((0|1)^* 00| (0|1)^* 11) (0|1)^*$

$0^* 1^* 2^*$

**Утверждение 5.**  $\forall \varepsilon$ -НКА  $\exists$  эквивалентный НКА без  $\varepsilon$  переходов

*Доказательство.* 1. Рассмотрим граф  $\varepsilon$ -переходов. Этому графу мы сделаем транзитивное замыкание. И добавим новые рёбра как  $\varepsilon$ -переходы в наш граф

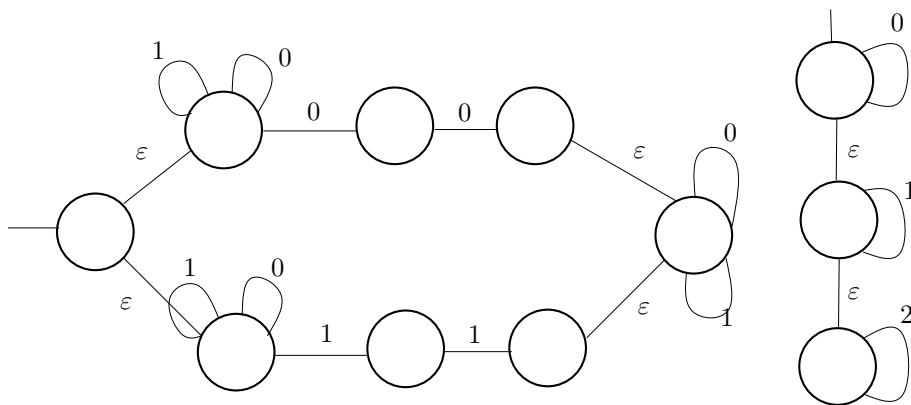


Рис. 1.10: epsauto

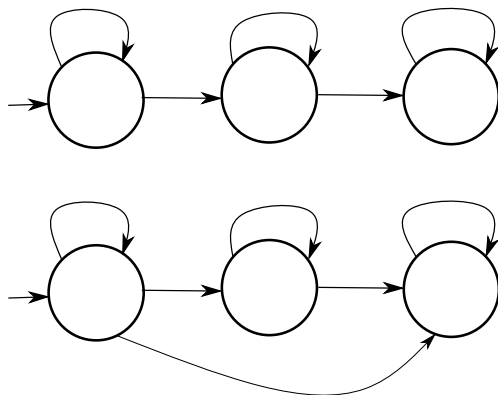


Рис. 1.11: epsgraph

Язык не поменялся. Вместо прохождения по новому переходу, можно делать  $n$  эpsilon-переходов в старом графе

2. Для каждой конструкции: Из  $p$  есть  $\epsilon$ -переход в  $q$  терминальный, сделаем  $p$  тоже терминальным

**Утверждение 6.** Если  $x$  Допускалось раньше,  $\iff$  допускается и сейчас. Последний переход не  $\epsilon$

3. Рассмотрим тройки вершин, что Из  $p$  есть  $\varepsilon$  переход в  $q$  откуда переход по  $c$  в  $r$ . Тогда добавим ребро из  $p$  в  $r$  по  $c$

**Утверждение 7.** Если слово можно допустить, то его можно допустить вообще не делая  $\varepsilon$ -переход

4. удалим все  $\varepsilon$ -переходы

■

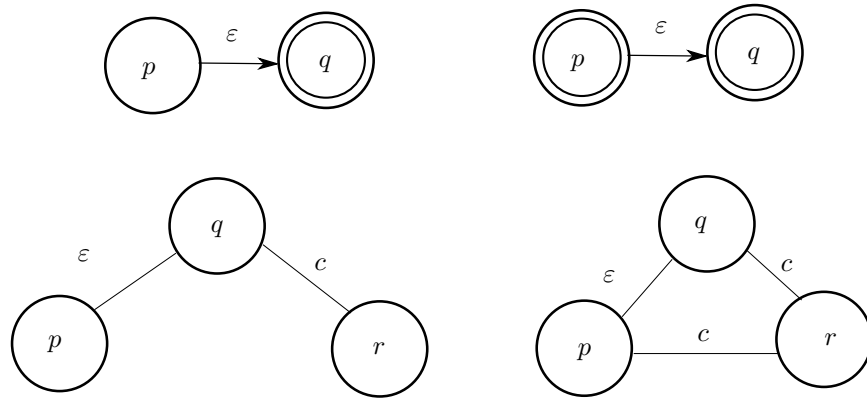


Рис. 1.12: proof-pic

**Утверждение 8.** Для любого Языка следующие три утверждения эквивалентны:

1. Можно построить ДКА
  2. Можно построить НКА
  3. Можно построить  $\varepsilon$ -НКА
- 2  $\implies$  1 – Томпсон  
3  $\implies$  1 –  $\varepsilon$ -замыкание

**Теорема 14** (Клини).  $\text{Reg} = \text{Aut}$

*Доказательство.*

$\text{Reg} \subseteq \text{Aut}$  Докажем по индекции, что  $\forall i \quad \text{Reg}_i \subseteq \text{Aut}$

Будем строить  $\varepsilon$ -НКА с одним терминальным состоянием

База:  $\emptyset$ . Стратовое и терминальное состояние и никаких переходов

$\varepsilon$  – одна  $\varepsilon$ -стрелка

$c$  – одна  $c$ -стрелка

Переход:  $\text{Reg}_i \subseteq \text{Aut} \implies \text{Reg}_{i+1} \subseteq \text{Aut}$

- $L = A \cup B$
- $L = AB$
- $L = A^*$

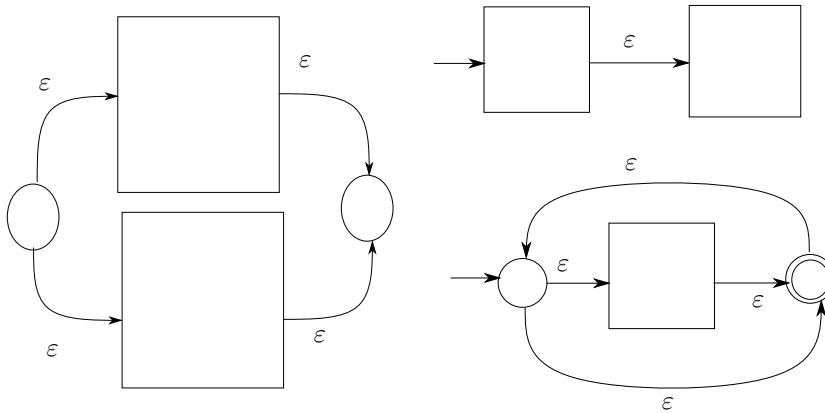


Рис. 1.13: move-klini

■

**Теорема 15** (Лемма о разрастании/накачке).  $L$  – регулярный

$\exists n > 0 \forall \omega : \omega \in L \quad |\omega| \geq n \quad \exists x, y, z : \quad \omega = xyz$

$y \neq \varepsilon \quad |xy| \leq n \quad \forall l \geq 0 \quad xy^l z \in L$

<TODOOTODOTODO>

**Замечание.** Формула, в которой чередуются блоки “существует” и “для любого” называется  $\sigma - \pi$ -формулой

Формула сверху это  $\sigma - 4$

На это можно смотреть на игру между двумя игроками. Существует ход одного, что для любого хода другого существует ход первого... Если утверждение верное, то выиграл “существует”, иначе выиграл “для любого”

**Пример.** Правильные скобочные последовательности

$$n \quad \omega = ( )^n \quad x = ( \quad y = ( \quad b > 0. z = (^{n-a-b})^n$$

$$k = 2 \quad (^{n+b})^n \notin L$$

*Доказательство.*  $L$  – регулярный язык.  $A$  – ДКА для  $L$

$\square$   $n$  – число состояний автомата  $A$

$\triangleleft w \in L \quad |w| \geq n$ . Раз оно из  $L$ , значит автомат его допускает

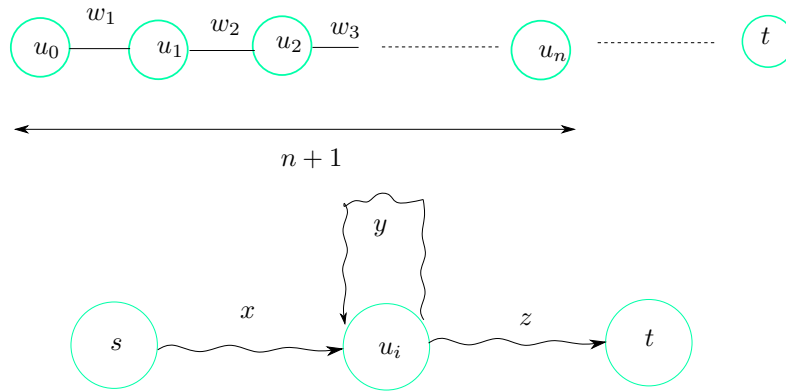


Рис. 1.14: dopusk

$$i \neq j \quad u_i = u_j \quad w[1 \dots i] = x \quad w[i+1 \dots j] = y \quad w[j+1 \dots |w|] = z \quad \blacksquare$$

**Пример.**  $0^a 1^a, a \geq 0$

$$n \quad 0^n 1^n$$

$$x = 0^a \quad y = 0^b \quad z = 0^{n-b-a} 1^n$$

$$k = 0$$

$$xy^k z = 0^{n-c} 1^n \notin L$$

**Замечание.** Просто ДКА для одного языка может быть несколько

Но у любого языка существует единственный ДКА с минимальным количеством состояний



## 1.11 Алгоритм Минимизации

ДКА  $A$ .  $u$  и  $v$  различим строкой  $S$ , если  $(x \in T \wedge y \notin T) \vee (x \notin T \wedge y \in T)$

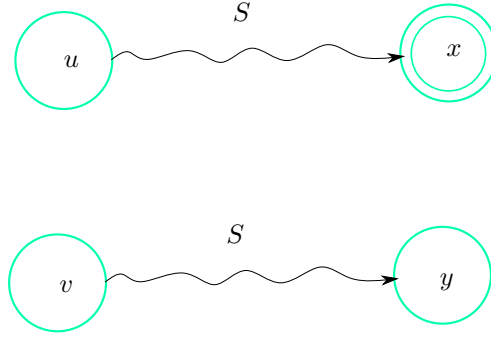


Рис. 1.15: razlich

$a \sim b$ , если они не различили никакой строкой

**Лемма 5.**  $\sim$  – отношение эквивалентности

*Доказательство.* Рефлексивность и Симметричность очевидны

■

**Лемма 6.**  $u \sim v \implies \delta(u, c) \sim \delta(v, c)$  для  $\forall c \in \Sigma$

*Доказательство.*  $\delta(u, c)$  и  $\delta(v, c)$  различимы  $S \implies u$  и  $v$  различимы  $cs$  ■

Алгоритм:

$$D_k = \{(u, v) \mid u \text{ и } v \text{ различимы } s, |s| \leq k\}$$

$$D_0 = \{(u, v) \mid u \in T \oplus v \in T\}$$

$$D_k = \{(u, v) \mid (u, v) \in D_{k-1} \text{ или } \exists x \in \Sigma : (\delta(u, x), \delta(v, x)) \in D_{k-1}\}$$

$$D_k = D_{k-1} \implies D_{k+1} = D_k \implies \forall \text{ пара различима в } D_k$$

Обход в ширину:

очередь  $Q$ , поместим  $D_0$  в  $Q$  и  $D_0$  в  $D$

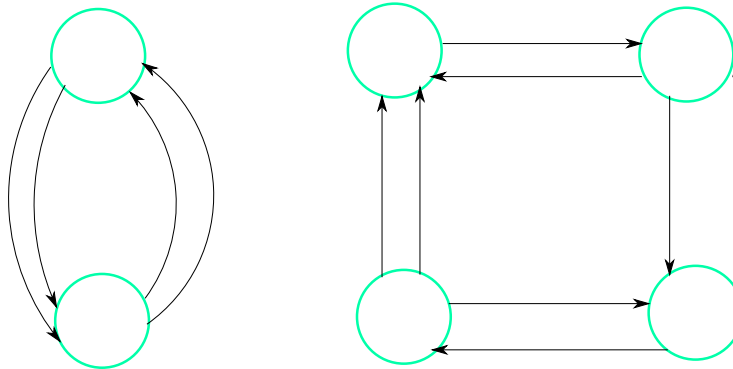


Рис. 1.16: minex

```

1 while not Q.empty()
2   (u, v) = Q.pop()
3   for c \in Sigma
4     for a \in In[u][c]
5       for b \in In[v][c]
6         if (a,b) !\in D
7           Q.push(a,b)
8           D.add(a,b)

```

while –  $n^2$   $|\Sigma| = \sigma$

форы суммарно –  $n^2$

Суммарно  $O(n^2 \cdot \sigma)$

**Теорема 16.**  $A$  – ДКА для  $L$ , не содержащий эквивалентных состояний, и любое состояние достижимо из  $S$ , тогда:

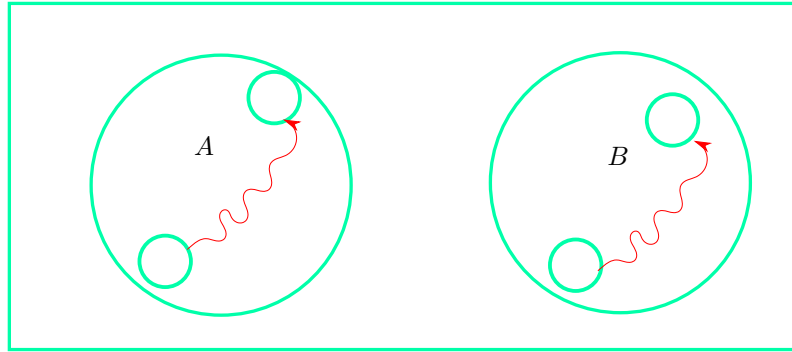
1.  $A$  – минимальный
2.  $A'$  – ДКА для  $L$ ,  $|Q'| = |Q|$ , тогда  $A \simeq A'$

Стартовые состояния эквивалентны. состояния, в которые можно прийти из них – тоже эквивалентны.

Есть алгоритм быстрой минимизации (фамилии, которую я не услышал), буит на след. неделе.

План

1. Другой алгоритм построения регулярного выражения по автомату



$A \cup B$

Рис. 1.17: union

(решаем системы линейных уравнений в регулярных выражениях)

2. Быстрый алгоритм минимизации (за  $n \log n$ )
3. Алгоритмический анализ свойств регулярных языков

## 1.12 Построение регулярки с помощью линала

$\alpha, \beta$  – регулярки

$L = \alpha L + \beta$  + здесь означает объединение или “или”

**Пример.**  $\alpha = aL + b$

решени –  $a^*b$

$\forall k \geq 0 a^k b \in L$

$$x \in L \quad x = b \begin{cases} \implies x \in a^*b \\ x \in aL \end{cases}$$

**Лемма 7.**  $\alpha^*b$  – решение

**Теорема 17.** Если  $\varepsilon \notin \alpha \implies \alpha^*\beta$  – единственное решение

Иначе  $L = \alpha^*b \cup N$  – решение для  $\forall N \subseteq \Sigma^*$

---

*Доказательство.* 1.  $\varepsilon \notin \alpha$

$$|x| = 0 \implies x \in L \implies x \in \alpha L + \beta \implies x \in \beta \implies x \in \alpha^* \beta$$

$$|y| < |x| \implies y \in \alpha^* \beta$$

$$x \in L \implies x \in \alpha L + \beta$$

$$(a) \ x \in \beta \implies x \in \alpha^* \beta$$

$$(b) \ x \in \alpha L \implies x = yz, y \in \alpha, z \in L$$

$$|y| \geq 1 \implies |z| < |z| \implies z \in \alpha^* \beta \implies x \in \alpha^* \beta$$

2.  $\varepsilon \in \alpha$

$$\beta \subseteq L \implies \alpha\beta \subseteq L \implies \dots \implies \alpha^* \beta \subseteq L$$

$$\forall N \quad \alpha N \supset N$$

■

**Задача 2.**  $X_1, X_2, \dots, X_n$  – неизвестные языки, которые связаны системой

$$\begin{cases} X_1 = \alpha_{11}X_1 + \alpha_{12}X_2 + \dots + \alpha_{1n}X_n + \beta_1 \\ X_2 = \alpha_{21}X_1 + \alpha_{22}X_2 + \dots + \alpha_{2n}X_n + \beta_2 \\ \vdots \\ X_n = \alpha_{n1}X_1 + \alpha_{n2}X_2 + \dots + \alpha_{nn}X_n + \beta_n \end{cases}$$

*Решение.* Потребуем  $\varepsilon \notin \alpha_{ij}$

Будем решать чем-то похожим на метод Гаусса.

Предположим, что  $X_1$  неизвестное, а остальное нам известно, тогда первое уравнение выглядит как

$$X_1 = \alpha_{11}^* (\alpha_{12}X_2 + \alpha_{13}X_3 + \dots + \alpha_{1n}X_n + \beta)$$

$$X_2 = \alpha_{21}\alpha_{11}^* (\alpha_{12}X_2 + \dots + \alpha_{1n}X_n + \beta) + \alpha_{22}X_2 + \dots + \beta_2$$

$$X_2 = (\alpha_{21}\alpha_{11}^*\alpha_{12} + \alpha_{22})^* ((\alpha_{21}\alpha_{11}^*\alpha_{13} + \alpha_{23})X_3 + \dots + (\alpha_{21}\alpha_{11}^*\beta_1 + \beta_2))$$

Так продолжаем до  $X_n$ . Кажется, что непохоже, но вообще похоже на прошлый алгоритм. ■

**Замечание.** Как это можно применить для автомата?

Рассмотрим автомат для чисел кратных трём

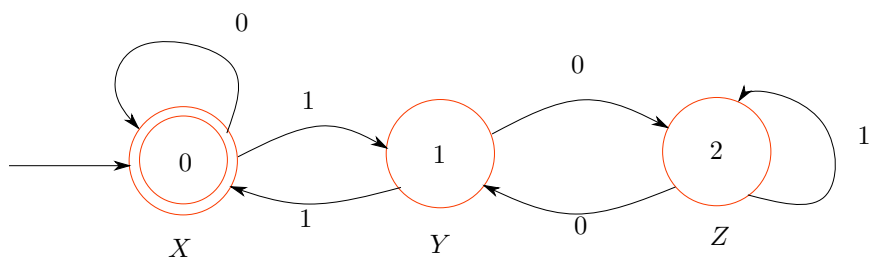


Рис. 1.18: kratno3

$$\begin{cases} X = 0X + 1Y + \varepsilon \\ Y = 0Z + 1X \\ Z = 1Z + 0Y \end{cases}$$

$$Z = 1^*0Y$$

$$Y = 01^*0Y + 1X$$

$$Y = (p_1^*0)^*1X$$

$$X = (0 + 1(01^*0)^*1)X + \varepsilon$$

$$X = (0 + 1(01^*0)^*1)^*$$

## 1.13 Алгоритмы анализа регулярных языков

---

**Теорема 18.** Пусть  $A, B$  – регулярные языки. Тогда такими также являются: (по определению  $A \cup B, AB, A^*$ )

- $A \cap B$
- $\overline{A}$
- $\varphi : \Sigma \rightarrow \Pi^* \quad \varphi^* : \Sigma^* \rightarrow \Pi^*$  – гомоморфизмы
 
$$\varphi^*(c_1 c_2 \dots c_n) = \varphi(c_1) \varphi(c_2) \dots \varphi(c_n)$$

$$\varphi L \Sigma^* \rightarrow \Pi^* \quad \varphi = \varphi^* \text{ (обозначим так)}$$

$$\varphi(A) \quad \varphi^{-1}(A) = \{x | \varphi(x) \in A\} \text{ – тоже регулярные}$$

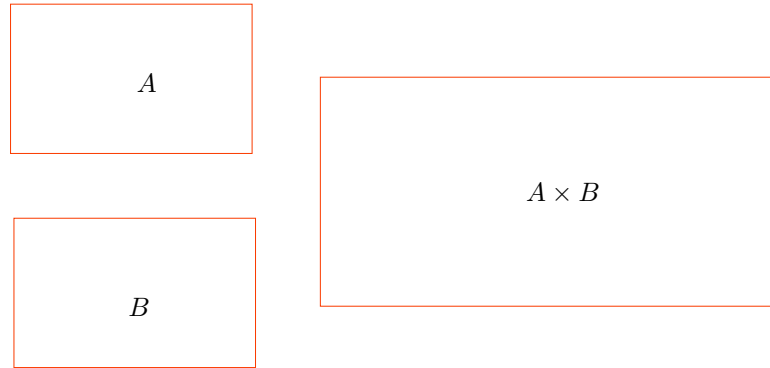


Рис. 1.19: preseh

*Доказательство.* Рассмотрим произведение автоматов.

Множество состояний – декартово произведение  $Q_{A \times b} = Q_A \times Q_b$

$$\delta(\langle u_a, u_b \rangle, c) = \langle \delta(u_a, c), \delta_B(u_b, c) \rangle$$

$$S = \langle S_a, S_b \rangle$$

$T = \{ \langle t_A, t_B \rangle \mid t_A \in T_A, t_B \in T_b \}$  – допускает, если оба автомата допускают.

Подобным образом можно сдать объединение, ксор, ... (см следствие)

$A$  – регулярный.  $\varphi^{-1}(A)$

$$\varphi : \Sigma \rightarrow \Pi^*$$

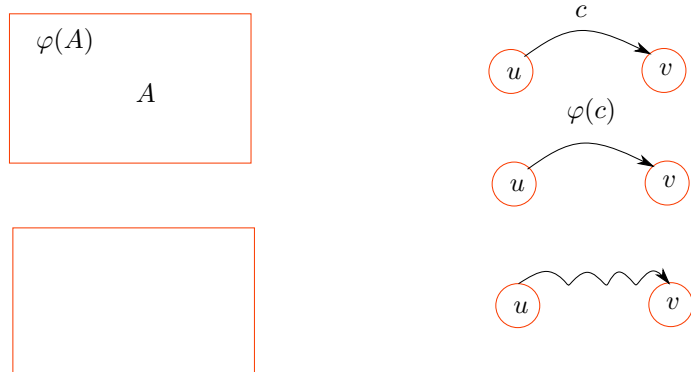


Рис. 1.20: homomор

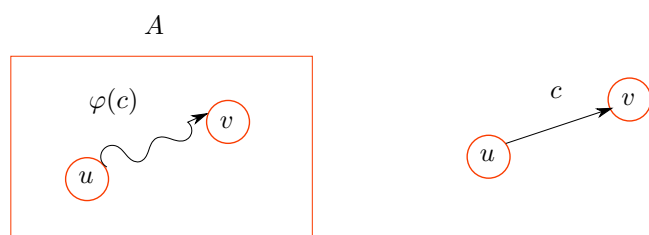


Рис. 1.21: обр-гомомор

■

**Следствие 1.**  $A_1, A_2, \dots, A_n$  – регулярные языки

$$f : \mathbb{B}^n \rightarrow \mathbb{B}$$

$$A = \{x \mid f(x \in A_1, x \in A_2, \dots, x \in A_n)\}$$

То  $A$  – регулярный. Доказательство не меняется, просто терминальными объявляем те, которые подходят под функцию

**Пример.**  $0^{2n}1^{2n}$  – нерегулярный

$$\varphi : \begin{cases} 0 \rightarrow 00 \\ 1 \rightarrow 11 \end{cases}$$

$$A = \varphi(0^N 1^n)$$

$$0^n 1^n = \varphi^{-1}(A)$$

## 1.14 Алгоритм Хопкрофта

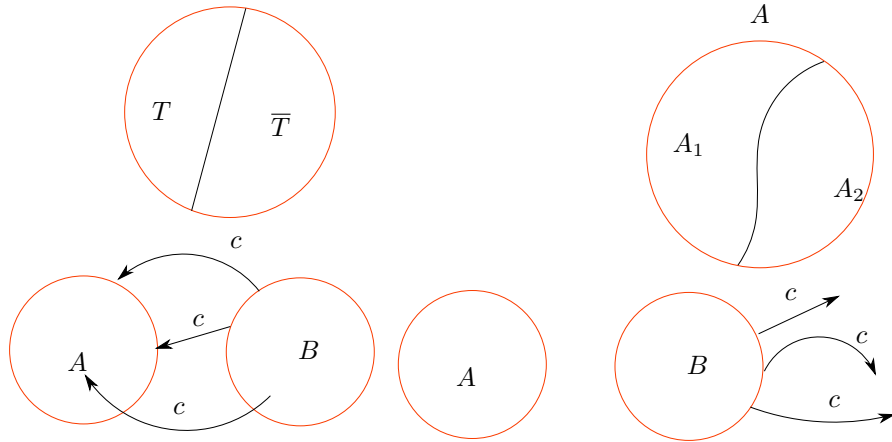


Рис. 1.22: classes

Хотим разбить на классы по различимости, чтоб бегать по ним за линию. Также нам потребуется список входящих рёбер по символу.

Пары состояний, в которых либо по символу все идёт из одного в другой, либо все ведут не в A.

Если есть и те, и те, разобьём их на ведущие в и не в.

Рассмотри Декеу, которая будет хранить классы, которые могут ещё повлиять на разделение других.



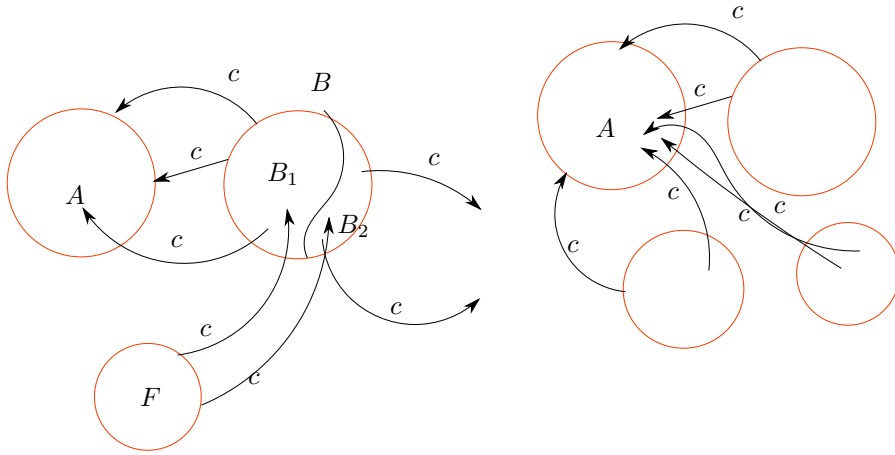


Рис. 1.23: nablud

$$\forall B : \exists u \in B \quad \delta(u, c) \in A \implies \forall u \in B \quad \delta(u, c) \in A$$

Если  $A$  – хороший, то он не будет влиять на разбиение других. Если нет, разбиваем все, что идут в него и делаем его хорошим.

В очереди будем хранить пары  $\langle A, c \rangle$  – состояние, символ, чтобы не разбивать сразу по всем символам (потому что это неудобно)

Если мы разбиваем  $B$ , то нужно положить  $\langle B_1, c \rangle \quad \langle B_2, c \rangle$

Хотим удалять, но удалять из очереди непоятно как. Но мы храним номер и одной половине оставляем старый, а другой даём новый. Так, если в очереди уже было состояние, то нужно добавить только по новому номеру

Каждая вершина просматривается  $n$  раз (столько может уменьшаться), рёбер  $n \cdot c$

Алгоритм делает всё за  $n^2 S + n^2$

Оптимизация 1:

$\triangleleft \langle B, d \rangle$  – хорошая. Оно разобьётся  $\langle B_1, d \rangle \quad \langle B_2, d \rangle$  – эти уже могли стать плохими. Но разбивать по обеим парам не нужно, т.к. разбивание по одной также сделает другую хорошей. Поэтому при разделении будем класть только ту, которая меньше

Если  $\langle B, d \rangle$  – плохой, то надо положить обе (добавим новую)

Теперь оно работает за  $n \log n S + n^2$

Оптимизация 2:

---

При разделении объявим новым классом тот, который указывает в  $C$ . Так у нас ... <пересмотреть лекцию и понять почему оно работает хорошо>

Дальше нам нужно уметь удалять из нашей структуры, можно либо связный список, либо hash-set

## 1.15 КС (контекстно-свободные) грамматики.

### Лекция 11 мая

TODO на пол пары

**Пример.**

$$\begin{aligned} A &\rightarrow \varepsilon \\ A &\rightarrow (A) \\ A &\rightarrow AA \end{aligned}$$

**Определение 28.** Грамматика называется право-линейной, если каждая правая часть задержит не более одного нетерминала и этот нетерминал находится на последнем месте (Все правила имеют вид  $A \rightarrow \alpha B$   $\alpha \in \Sigma^*$   $A \rightarrow \alpha$ )

**Теорема 19.** Язык  $A$  регулярный  $\iff A$  задаётся праволинейной грамматикой.

*Доказательство.*

$\implies$   $\triangleleft$  ДКА для языка  $A$ .  $\langle \Sigma, Q, S, T, \delta \rangle$

$N = Q$   $S = S$   $\delta$  – правила:

Если из  $X$  переходим в  $Y$  по символу  $c$ , то добавляем  $X \rightarrow cY$

Для терминального состояния  $T$  добавляем правило  $T \rightarrow \varepsilon$

По построению видно, что всё правильно  $=)$

$$\langle s, xy \rangle \vdash^k \langle U, y \rangle$$

$$S \rightarrow^k xU$$

$\Leftarrow$  Строим НКА с  $\varepsilon$ -переходами

$\langle \Sigma, N, S, P \rangle$  – грамматика

$$Q = N \cup \{T\} \quad S = S \quad \delta :$$

---


$$A \rightarrow \alpha B$$

1.  $\alpha = \varepsilon$   $A \rightarrow B$  по  $\varepsilon$
2.  $\alpha = c$  переход по  $c$
3.  $\alpha = c_1 c_2 \dots c_k$  промежуточные вспомогательные состояния по одному символу

$A \rightarrow \alpha$  – сделать терминальное состояние.

■

**Следствие 2.**  $Reg \subseteq CF$   $Reg \neq CF$

**Замечание.** Почему контекстно-свободные. Потому что нет контекста. Можно дальше ещё обобщить

Иерархия Хосмкого. У нас есть некоторое неравноправие, слева всегда стоит только один символ. Давайте разрешим там тоже несколько символов.

Формальная грамматика нулевого класса – как грамматика, только слева стоит произвольная строка  $\Gamma = \langle \Sigma, N, S, P \subseteq N^+ \times (N \cup \Sigma)^* \rangle$

$0^n 1^n 2^n$  – не задать даже  $CF$  грамматикой, но можно формальной грамматикой нулевого класса

$$\begin{aligned}
 S &\rightarrow ZT \\
 T &\rightarrow ABCT \\
 T &\rightarrow \varepsilon \\
 BA &\rightarrow AB \\
 CB &\rightarrow BC \\
 CA &\rightarrow AC \\
 ZA &\rightarrow 0Z \\
 Z &\rightarrow Y \\
 YB &\rightarrow 1Y \\
 Y &\rightarrow X \\
 XC &\rightarrow 3X \\
 X &\rightarrow \varepsilon
 \end{aligned}$$

Грамматики нулевого класса задают все языки, порождаемые Машины Тьюринга

CG (Chomsky)

$CG_1$  – контекстно зависимые грамматики. Правила обладают контекстом  $\xi A \eta \rightarrow \xi \alpha \eta \quad \xi, \eta \in (\Sigma \cup N)^*$  КЗГ

$CG_2 = CFG$

$CG_3$  = праволинейные грамматики

**Определение 29.** Неукорачивающая грамматика.  $\alpha \rightarrow \beta \quad |\alpha| \leq |\beta|$   
 Грамматику выше можно сделать такой. Некоторые нет.

## 1.16 Выводиться ли слово в заданной КС

**Определение 30.** Г в НФХ (нормальная форма Хомского)

$A \rightarrow BC \quad A \rightarrow c$

(а также  $S \rightarrow \varepsilon$ , тогда  $S$  не встречается в правых частях)

1. В правых частях только  $N$ , кроме правил  $A \rightarrow c$

$cN_c \quad N_c \rightarrow c \quad A \rightarrow \alpha c \beta \quad A \rightarrow \alpha N c \beta$

- $A \rightarrow \varepsilon$  – эpsilon правила
- $A \rightarrow B$  – цепные правила
- $A \rightarrow BC$  – хорошие правила, мы так любим, остальные враги
- $A \rightarrow N_1 N_2 \dots N_k, k \geq 3$  – длинные правила

**Пример.**  $S \rightarrow (S)S \quad S \rightarrow \varepsilon$

$S \rightarrow ASBS \quad A \rightarrow ( \quad B \rightarrow ) \quad S \rightarrow \varepsilon$

Легче всего избавиться от длинных правил.

$A \rightarrow N_1 N_2 \dots N_k$

$A \rightarrow N_1 Y_1 \quad Y_1 \rightarrow N_2 Y_2 \quad \dots \quad Y_{k-2} \rightarrow N_{k-1} N_k$

**Пример.**  $S \rightarrow AX \quad X \rightarrow SY \quad Y \rightarrow BS \quad A \rightarrow C \quad B \rightarrow ) \quad S \rightarrow \varepsilon$

Удаление  $\varepsilon$ -переходов:

(а) Найдём  $\varepsilon$ -переходы.  $A \Rightarrow^* \varepsilon$

$A \rightarrow \varepsilon \quad B \rightarrow AA \quad C \rightarrow AB$  – все они  $\varepsilon$ -порождающие

```

1         while changes:
2             for A -> alpha:
3                 Если все нетерминальные в alpha --
4                 epsilonпорождающие-
                    то пометить A как epsilon
                    порождающий-

```

$B \Rightarrow^k \varepsilon$   $k \rightarrow \min$  алгоритм не пометил  $B$  как  $\varepsilon$ -порождающую

$B \Rightarrow C \Rightarrow \dots \Rightarrow \varepsilon$

$B \Rightarrow CD \Rightarrow \dots \Rightarrow \varepsilon$

$C$  и  $D$  – были порождены на  $k-1$  шаге, значит и  $B$  тоже должен был быть помечен

(b)  $A \Rightarrow^* \varepsilon$

$B \Rightarrow AC$ . Добавим правило  $B \Rightarrow C$

$B \Rightarrow CA$ . Добавим  $B \Rightarrow C$

Если  $S$  –  $\varepsilon$ -порождающий, заведём новое стартовое состояние и правило  $S' \rightarrow S$   $S' \rightarrow \varepsilon$

**Пример.**  $S \rightarrow AX$   $X \rightarrow Sy$   $X \rightarrow Y$   $Y \rightarrow BS$   $Y \rightarrow$   
 $B$   $A \rightarrow (B \rightarrow)$   $S' \rightarrow \varepsilon$   $S' \rightarrow S$

Граф цепных правил

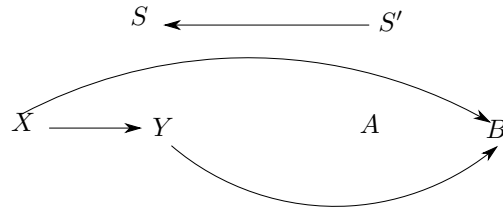


Рис. 1.24: chain-graph

Транзитивное замыкание  $A \rightarrow B$   $B \rightarrow \alpha$   $A \rightarrow \alpha$

**Пример.**  $S \rightarrow AX$   $X \rightarrow SY$   $X \rightarrow BS$   $X \rightarrow (Y \rightarrow BS \ Y \rightarrow)$   $A \rightarrow$   
 $(B \rightarrow)$   $S' \rightarrow \varepsilon$   $S' \rightarrow AX$

()()

## 1.17 Алгоритм Кока-Янгера-Касами (КЯК) Cock, Yanger, Kasami СΥК

Принимает на вход грамматику в НФХ и слово  $x$   $len(x) = n$   $x[0 \dots n-1]$

$d_a[l][r]$  – можно ли из нетерминала  $a$  породить фрагмент нашего слова  $x[l \dots r-1]$

$$d_A[l][r] = \bigvee_{A \rightarrow BC} \bigvee_{k=l+1}^{r-1} (d_B[l][k] \wedge d_C[k][r]) \quad r-l > 1$$

$$d_A[l][l+1] = \begin{cases} 1 & , A \rightarrow x[l] \in \Gamma \\ 0 & , \text{иначе} \end{cases}$$

$$ans = d_s[0][n]$$

```

1   for len = 2 ... n:
2       for l = 0 ... n-len:
3           r = l + len:
4               ...

```

Итоговая сложность  $|\Gamma| n^3$

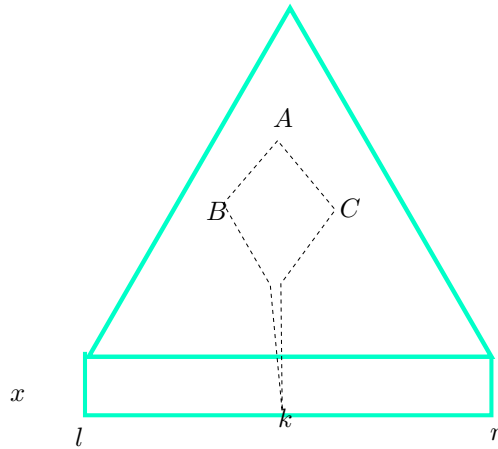


Рис. 1.25: cyk-algo

**Пример.**  $(0|1)^*00(0|1)^*$  – слова с двумя нулями подряд

Можно написать автомат. Он распознаёт строку

Грмматматика  $S \rightarrow (S)S \quad S \rightarrow \varepsilon$

С одной стороны мы говорим как порождать слова. С другой стороны с помощью алгоритма выше можно проверять выводиться ли слово в КС-грамматике.

**Пример.** Грамматика для палиндромов

$S \rightarrow aSa \quad S \rightarrow bSb \quad S \rightarrow \varepsilon$ . Проверить что слово палиндром можно также например стэком, дойдя до середины

## 1.18 Автомат с магазином памяти. МП-автомат, Стековая машина, Автомат со стэком, Push-Down-Automaton (PDA)

Придётся делать НКА, потому что мы не можем чётко знать что нужно хранить (пример слова асканчиваюиеся на 001, он не знает когда нужно выходить на конец, а просто рассматривает возможность всегда когда она есть)

В стэке изначально находится маркер дна. Его не надо вынимать или копировать.

В рамках перехода. включается два параметра: символ на которым мы сейчас  $c$  (указатель на ленте) и верхний символ в стэке  $A$ .

При переходе мы можем изменить стэк:  $c, A/\alpha$  – по символу  $c$  мы переходим, убираем  $A$  и кладём в стэк  $\alpha$

**Пример.** Будем хранить баланс в унарной системе

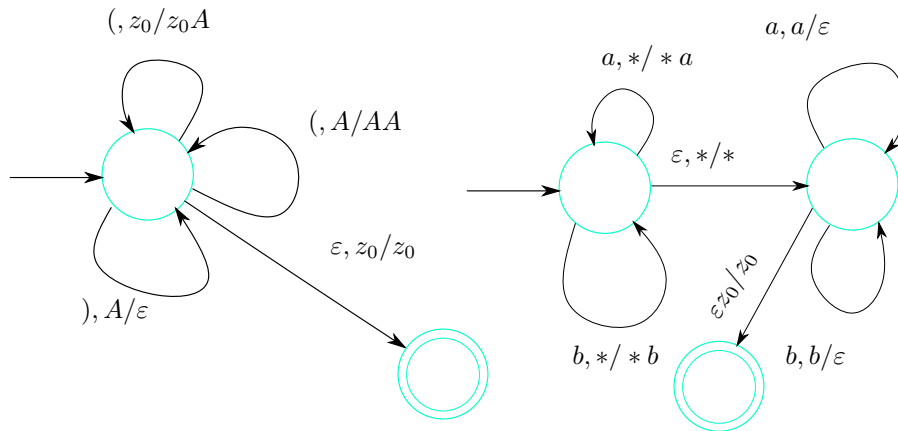


Рис. 1.26: brack-stack

---

Входной Алфавит  $\Sigma$

Стэковый алфавит  $\Pi$

Состояния  $Q$

Стартовое состояние  $S \in Q$

Допускающее состояние  $T \in Q$

$\delta : Q \times (\Sigma \cup \varepsilon) \times \Pi \rightarrow \mathcal{P}_{<+\infty}(Q \times \Pi^*)$  – Powerset  $<+\infty$  – семейство конечных пожмножеств.

**Теорема 20.**  $L$  – контекстно-свободный  $\iff L$  распознаётся МП-автоматом

### 1.19 МП-автоматом с допуском по пустому стэку

**Определение 31.** Нету терминальных состояний, но теперь можем снимать маркер дна.

Допуск, если полностью опустошили стэк.

**Определение 32** (операция переходит за один шаг).  $\langle q, x, \gamma \rangle \vdash \langle r, y, p \rangle$   
состояние-вход(слово)-стэк

**Определение 33** (МП-автомат-язык).  $L(A) = \{x \mid \langle s, x, z_0 \rangle \vdash^* \langle t, \varepsilon, \varepsilon \rangle\}$

**Лемма 8.** Допуск по допускающему состоянию, допуск по пустому стэку

*Доказательство.* ■

$KC \implies$  МП-автомат. строим МП-автомат с допуском по пустому стэку (ДПС)

$\Sigma, \Pi = \Sigma \cup N, Q = \{st\}, S = st, z_0 = S, \delta$

$A \rightarrow \alpha \implies \varepsilon, A/\alpha$

$A \rightarrow c \implies c, c/\varepsilon$

$S \implies \xi = xA\zeta \implies x\alpha\zeta$

$\langle st, xy, S \rangle \vdash^* \langle st, y, \alpha \rangle$  ■



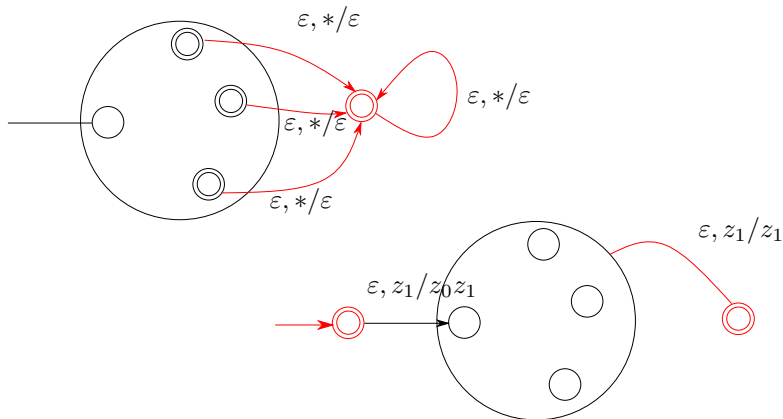


Рис. 1.27: ekvimp

**Определение 34** (LL(k) грамматики). Если можно подсмотрев на  $k$  символов точно определить какое правило применить (детерминированность)

МП-автомат  $\implies$  КСГ. Автомат  $\Sigma, \Pi, Q, s \in Q, z_0 \in \Pi, \delta$

КСГ  $\Sigma, Q \times P \times Q[pAq]$ ,

состояние  $P$  символ  $A$  перейдём в  $r$  с состояни

$L[pAq] = \{x \mid \langle p, x, A \rangle \vdash^* \langle q, \varepsilon, \varepsilon \rangle\}$

$S \rightarrow [sz_0q_1] \quad S \rightarrow [sz_0q_2] \quad \dots \quad S \rightarrow [sz_0q_k]$

$p \rightarrow r \quad c, A/\alpha$

$c \in \Sigma \cup \{\varepsilon\}$

1.  $\alpha = \varepsilon$  должно быть  $q = r$

$[pAr] \rightarrow c$

2.  $|\alpha| = 1 \quad \alpha = B$

$[pAq] \rightarrow c[rBq]$

3.  $|\alpha| \geq 2 \quad \alpha = B_1B_2 \dots B_k, k \geq 2$

$\forall u_1, u_2, \dots, u_{k-1} \in Q \quad (|Q|^{k-1})$

$[pAq] \rightarrow c[rB_1u_1][u_1B_2u_2] \dots [u_{k-1}B_kq]$

■

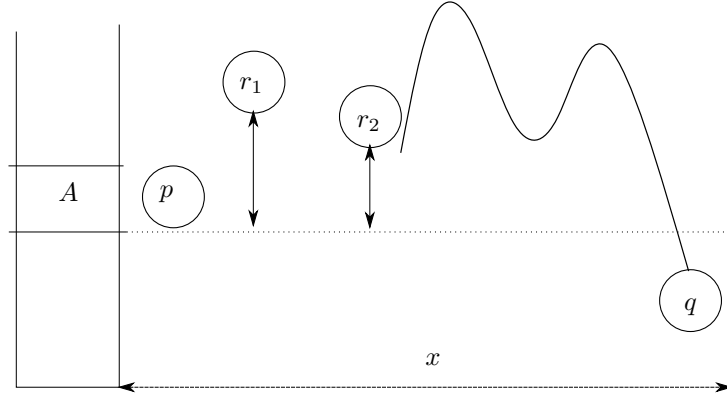


Рис. 1.28: klast-snimat

МП-автомат  $A \mapsto \Gamma \xrightarrow{\text{НФХ}} \Gamma' \rightarrow A'$

$A'$  кладёт в стек не более двух символов и эквивалентен  $A$ !

**Пример.**  $\Sigma, x_1, x_2, \dots, x_n$  – множество слов

$$\Sigma' = \Sigma \cup \{1, \dots, n\}$$

$$L : S \rightarrow \varepsilon, S \rightarrow x_i S i$$

$$3121x_1x_2x_1x_3$$

$L$  – КС

**Теорема 21** (Лемма о разрастании (накачке boost) для КС).  $L$  – КС-язык

$$\begin{aligned} \exists n \forall w \in L, |w| \geq n \quad \exists u, v, x, y, z : w = uvxyz \quad |vxy| \leq n \quad vy \neq \varepsilon \\ \forall k \geq 0 \quad uv^kxy^kz \in L \end{aligned}$$

*Доказательство.*  $\sqsupset \Gamma$  – КСГ в НФХ для  $L$

$\Gamma$  содержит  $m$  нетерминалов

выберем  $n = 2^m$

**Лемма 9.**  $\sqsupset w \in L \quad |w| \geq n$

Тогда в дереве разбора  $w$  существует такой нетерминал  $A$ , что у него в собственном поддереве тоже есть нетерминал  $A$

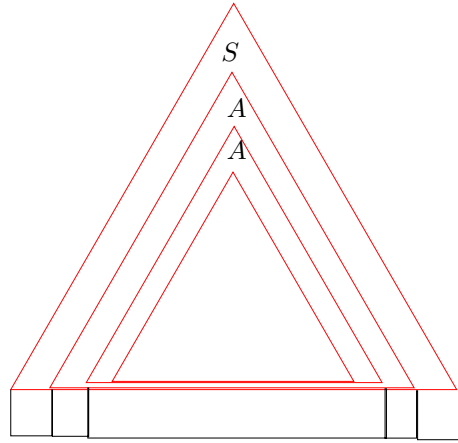


Рис. 1.29: prostaya-kartinka

*Доказательство.* Допустим нет, тогда , значит суммарное количество листьев не более  $2^{m-1}?! (n = 2^m)$  ■

Выберем пару  $A - - - A$ , у которой верхний нетерминал самый глубокий. В её поддереве таких пар нет (мы взяли самую глубокую), поэтому высота дерева в рёбрах не превышает  $m$

Крона этого поддерева  $|t| \leq 2^h = 2^m = n$

То, что справа и слева от поддерева  $A$  по листья назовём  $u$  и  $z$

Внутри по поддереву нижнего  $A$  Разделим на  $vxy$

$|vxy| \leq n$ , как и требовалось

Т.к. у верхнего  $A$  есть два сына, нижнее  $A$  лежит в либо правом, либо левом поддереве. Значит либо  $v$  либо  $y$  содержат хотя бы один символ (от другого сына верхнего  $A$ ) ■

**Теорема 22.** Язык  $0^n 1^n 2^n$  – НЕ КС

*Доказательство.*  $\leq n \quad w = 0^n 1^n 2^n$

$|vxy| \leq n$ , соответственно не может содержать одновременно 0 и 2

$k = 2 \quad uv^2xy^2z$

длина увеличилось, либо количество 0  $n$ , либо количество 2  $n$ , значит их не треть, слово не из языка. ■

---

**Пример.**  $L = 0^n 1^n 2^m$

$S \rightarrow AB \quad A \rightarrow 0A_1 \quad A \rightarrow \varepsilon \quad B \rightarrow 2B \quad B \rightarrow \varepsilon$

$L = 0^n 1^m 2^m$  – тоже КС

$L_1 \cap L_2 = 0^n 1^n 2^n$  – Не КС

**Замечание.** Автомат с двумя стэками распознаёт вообще любые языки

**Теорема 23.**  $L \in CF \quad L_2 \in Reg \implies L_1 \cap L_2 \in CF$  (можно рассмотреть произведение обычного автомата и автомата со стэком)

**Теорема 24.** Язык тандемных повторов на односимвольном алфавите не является КС.

$\leq n \quad 0^n 1^n 0^n 1^n$

Если  $vxu$  целиком какой-то блок, берём  $k = 1$  и доказано

Если два блока: Если не посередине, то берём  $k = 2$  и две не равные части получаются

Если посередине  $0^a 1^b 0^n 1^n$

**Замечание.** Существуют КС языки, такие, что дополнение не КС