Дискретная математика. Теория

Александр Сергеев

1 Дискретная теория вероятности

1.1 Введение

Определение
Дискретное вероятностное пространство – пара (Ω, p),
где Ω - не более чем счетное множество элементарных исходов
$p : \Omega \to [0,1], \sum\limits_{\omega \in \Omega} p(\omega) = 1$

Определение
Событие – $A \subset \Omega$
Вероятность события – $P(A) = \sum\limits_{a \in A} p(a)$

Определение
События A и B *независимые*, если $P(A \cap B) = P(A)P(B)$

$P(B|A) = \dfrac{P(A \cap B)}{P(A)}$ - вероятность B при условии A

Лемма
Если A и B независимы, то $P(B|A) = P(B)$

Определение
Пусть $(\Omega_1, p_1), (\Omega_2, p_2)$ - независимые ДВП
Тогда их произведение $(\Omega = \Omega_1 \times \Omega_2, p(\omega_1 \in \Omega_1, \omega_2 \in \Omega_2) = p_1(\omega_1)p_2(\omega_2))$

Определение
$A_1, A_2, \ldots, A_n$ - *независимы в совокупности*,
если $\forall I \subset \{1 \ldots n\}$ $P(\bigcap\limits_{i \in I} A_i) = \prod\limits_{i \in I} P(A_i)$

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

Формула Байеса
$P(A_i|B) = \dfrac{P(B|A_i)P(A_i)}{P(B)} = \dfrac{P(B|A_i)P(A_i)}{\sum_{j=1}^{n} P(B|A_j)P(A_j)}$

Оффтоп
Рассмотрим пример: выкинули два честных кубика
Заметим, что возможно построить две математические модели:

1. Результаты - упорядоченная пара $\langle x, y \rangle$
   Тогда $|\Omega| = 36, p(w) = \dfrac{1}{36}$
2. Результаты - неупорядоченная пара $[x, y]$
   Тогда $|\Omega| = 21, p([x, x]) = \dfrac{1}{36}, p([x, y \neq x]) = \dfrac{1}{18}$

Заметим, что для запросов, не содержащих информацию об упорядоченности, результат не зависит от построенной модели

1.2 Случайные величины

Определение
Случайной величиной называется функция $\xi : \Omega \to \mathbb{R}$
Примеры случайных величин

1. Пусть кинули 10 монет. Построим случайную величину – количество выпавших орлов: $\xi(b_1, \ldots, b_n) = b_1 + \ldots + b_n$
2. Пусть кинули n кубиков. Построим случайную величину – среднее значение: $\xi(v_1, \ldots, v_n) = \dfrac{v_1 + \ldots + v_n}{n}$
3. Пусть n студентов приходят на лекцию с вероятностями $p_1, \ldots, p_n$. Построим случайную величину - количество студентов на лекции: $\xi(s_1, \ldots, s_n) = \sum_{i=1}^{n} s_i$

Заметим, что у этой случайной величины неравномерное распределение вероятностей: $p(s_1, \ldots, s_n) = \prod_{i=1}^{n} \begin{cases} p_i, & s_i = 1 \\ 1 - p_i, & s_i = 0 \end{cases}$

Давайте внализировать события через их случайные величины
Заметим, что уравнение $\xi = 3$ задает событие $\{w : \xi(w) = 3\}$(аналогично и другие предикаты с ξ задают события)

Определение

$f_\xi : \mathbb{R} \to \mathbb{R}, f_\xi(a) = P(\xi = a)$ – дискретная плотность распределения
$F_\xi : \mathbb{R} \to \mathbb{R}, f_\xi(a) = P(\xi \le a)$ – функция распределения

Определение

Пусть ξ – случайная величина
$E_f = \sum_{\omega \in \Omega} \xi(\omega)p(\omega)$ – математическое ожидание
$E_\xi = \sum_{\omega \in \Omega} \xi(\omega)p(\omega) = \sum_a \sum_{\omega:\xi(\omega)=a} = \sum_a a \sum_{\omega:\xi(\omega)=a} p(\omega) = \sum_a aP(\xi = a) = \sum_a af_\xi(a)$

Определение

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

Свойства математического ожидания

1. $E(c\xi) = cE_\xi$
2. $E(\xi + \eta) = E_\xi + E_\eta$ (даже для зависимых величин)
3. Для независимых ξ, η $E(\xi\eta) = E(\xi)E(\eta)$
   Доказательство
   $E(\xi\eta) = \sum_a aP(\xi\omega = a) = \sum_x \sum_y xyP(\xi = x \wedge \omega = y) = \sum_x x \sum_y yP(\xi = x \wedge \omega = y) = \sum_x x \sum_y yP(\xi = x)P(\omega = y) = \sum_x xP(\xi = x) \sum_y yP(\omega = y) = E_\xi E_\omega$
4. $E(\xi - E_\xi) = E\xi - EE\xi = E\xi - E\xi = 0$
5. $D_\xi = E((\xi - E\xi)^2) = E(\xi^2 - 2\xi E\xi + (E\xi)^2) = E(\xi^2) - E(2\xi E\xi) + E((E\xi)^2) = E(\xi^2) - (E\xi)^2$

Определение
ξ, η независимы, если $\forall\, a, b$ события $\xi = a$ и $\eta = b$ независимы
Для непрерывных величин вместо $=$ *берем* $\leq$

Пример 1
Бросаем два кубика
$\xi = v_1 + v_2$
$E_\xi = 7$
Пример 2
Бросаем кубик
$\xi = up + down$
$E_\xi = 7$
Пример 3
Ω – перестановки n элементов
$p(\sigma) = \dfrac{1}{n!}$
$\xi(\sigma) = |\{i : \sigma_i = i\}|$
Утверждается, что $E_\xi = 1$
Посчитать это через подсчет случаев сложно
Несмотря на это, мы можем посчитать матожидание
Пусть $\xi_i = (\sigma_i = i)$
$\xi = \xi_1 + \ldots + \xi_n$
$E_{\xi_i} = P(\xi_i = 1) = \dfrac{(n-1)!}{n!} = \dfrac{1}{n}$
Отсюда $\xi = n\dfrac{1}{n} = 1$

Свойства дисперсии $D(c\xi) = c^2 D(\xi)$
Дисперсия не линейна

Теорема
$D(\xi + \eta) = D(\xi) + D(\eta)$
Доказательство
$D(\xi+\eta) = E(\xi+\eta)^2 - E((\xi+\eta)^2) = E(\xi^2 + 2\xi\eta + \eta^2) - (E\xi)^2 - 2E\xi E\eta - (E\eta)^2 = E\xi^2 + 2E\xi E\eta + E\eta^2 - (E\xi)^2 - 2E\xi E\eta - (E\eta)^2 = E\xi^2 - (E\xi)^2 + E\eta^2 - (E\eta)^2 = D(\xi) + D(\eta)$
Следствие
$\xi_1, \ldots, \xi_n$ – одинаково распределенные независимые случайные величины

$\xi = \frac{1}{n}\sum_{i=1}^{n}\xi_i$

$E_\xi = E_{\xi_i}, D_\xi = \frac{1}{n}D_{\xi_i}$

Определение

$\sigma = \sqrt{D_\xi}$ – среднеквадратичное отклонение

1.3 Хвостовые неравенства

Неравенство Маркова

Пусть $\xi \geq 0, E_\xi > 0$

Оценим $P(\xi \geq cE_\xi) \leq \frac{1}{c}$

Доказательство

$P(\xi \geq cE_\xi) = \sum\limits_{\substack{\omega \\ \xi(\omega)\geq cE_\xi}} p(\omega)$

$E_\xi = \sum\limits_{\omega} p(\omega)\xi(\omega) = \sum\limits_{\substack{\omega \\ \xi(\omega)\geq cE_\xi}} p(\omega)\xi(\omega) + \sum\limits_{\substack{\omega \\ \xi(\omega)< cE_\xi}} p(\omega)\xi(\omega) \geq \sum\limits_{\substack{\omega \\ \xi(\omega)\geq cE_\xi}} p(\omega)\xi(\omega) \geq$

$cE_\xi \sum\limits_{\substack{\omega \\ \xi(\omega)\geq cE_\xi}} p(\omega) = cE_\xi P(\xi \geq cE_\xi)$

$1 \geq cP(\xi \geq cE_\xi)$

$P(\xi \geq cE_\xi) \leq \frac{1}{c}$

Неравенство Чебышева

$P(|\xi - E_\xi| \geq c\sqrt{D_\xi}) \leq \frac{1}{c^2}$ – относительная форма неравенства Чебышева

Доказательство

Возьмем $\eta = (\xi - E_\xi)^2$

Неравентсво Чебышева (ver. 2)

$c := \frac{a}{\sqrt{D_\xi}}$

$P(|\xi - E_\xi| \geq a) \leq \frac{D_\xi}{a^2}$ – абсолютная форма неравенства Чебышева

Пример

Возьмем честную монету

$E_\xi = \dfrac{1}{2}$
$D_\xi = \dfrac{1}{4}$
$D_\xi = E_\xi - (E_\xi)^2$
$P(|\xi - E_\xi| \geq \dfrac{1}{2}) \leq 1$
$P(|\xi - E_\xi| \geq 1) \leq \dfrac{1}{4}$ (на самом деле 0)
Видим, что оценка сверху неточная

Пример 2
$\xi_1, \ldots, \xi_n$ – одинаково распределенные независимые случайные величины
$\xi = \dfrac{1}{n}\sum\limits_{i=1}^{n}\xi_i$
$P(|\xi - E_\xi| \geq \varepsilon) \leq \dfrac{D_\xi}{\varepsilon^2}$
$P(|\xi - E_\xi| \geq \varepsilon) \leq \dfrac{D_{\xi_i}}{n\varepsilon^2}$
Пусть мы хотим не попадать в ε-окрестность с вероятностью не более δ (вероятность промаха)
$P(|\xi - E_\xi| < \varepsilon) > 1 - \delta$
$P(|\xi - E_\xi| \geq \varepsilon) \leq \delta$
Тогда $\dfrac{D_{\xi_i}}{n\varepsilon^2} \leq \delta$
$n \geq \dfrac{D_{\xi_i}}{\varepsilon^2\delta} \sim \dfrac{1}{\varepsilon^2\delta}$
Граница Чернова для монеты Бернулли
$P(\xi \geq (1+\delta)np) \leq e^{-\frac{\delta^2}{2+\delta}np}$
$P(\xi \leq (1-\delta)np) \leq e^{-\frac{\delta^2}{2}np}$
Доказательство
Доказательства не будет, жди теорвер

1.4 Введение в информатику

Определение
информация = -неопределенность (по Шеннону)

Рассмотрим модель случайного источника
Пусть есть вероятностное пространство Ω и распределение p
Получая событие ω, мы получаем информацию о том, что оно произошло
Определим, сколько информации мы получаем
Заметим, что оно не зависит от Ω
Пусть $H(p_1, p_2, \ldots)$ – количество информации в зависимости от вероятностей событий
H удовлетворяет следующим свойствам:

1. Для любого числа n $H(p_1, \ldots, p_n)$ – непрерывно
(т.к. при малом изменении вероятностей количество информации мало изменяется)

2. $H(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}) = h(n)$
$h(n) \uparrow$ – (т.к. чем больше вариантов, тем больше информации)

3. Аддитивность
Пусть $\Omega \subset \mathbb{R}^2$ – множество пар
$A_a = \{(a, *) \in \Omega\}$
$P(A_a) = p_a$
$P(\{(a, b)\}|A_a) = q_{a,b}$
$p(\{(a, b)\}) = p_a q_{a,b}$
Тогда $H(p_1 q_{1,1}, \ldots, p_1 q_{1,k_1}, \ldots, p_n q_{n,1}, \ldots, p_n q_{n,k_n}) = H(p_1, \ldots, p_n) + \sum\limits_i p_i H(q_{i,1}, \ldots, q_{i,k_i})$

Рассмотрим случай с равными вероятностями
$h(nm) = h(n) + \sum\limits_{i=1}^{n} \frac{1}{n} h(m) = h(n) + h(m)$
Лемма
$h(n) = c \log_2(n)$
Традиционно $c = h(2)$ – бит
Доказательство
$h(2^k) = kh(2) = ck$
Рассмотрим $n^t, n, t \in \mathbb{N}$
Пусть $2^k \leq n^t < 2^{k+1}$

Тогда $h(2^k) \le h(n^t) \le h(2^{k+1})$
$ck \le th(n) \le c(k+1)$
$\frac{ck}{t} \le h(n) \le \frac{c(k+1)}{t}$
$k \le t\log_2(n) \le k+1$
$\frac{ck}{t} \le c\log_2(n) \le \frac{c(k+1)}{t}$
$|h(n) - c\log_2(n)| \le \frac{c}{t}$ – при всех t
Отсюда $h(n) = c\log_2(n)$

"Разберемся"с H
Начнем с $p_i, q_i \in \mathbb{Q}$
Пусть $p_i = \frac{a_i}{b}$
$k_i = a_i, q_{ij} = \frac{1}{a_i}$
Отсюда $H(\frac{1}{b}, \ldots, \frac{1}{b}) = H(p_1, \ldots, p_n) + \sum_{i=1}^{n} p_i H(\frac{1}{a_i}, \ldots, \frac{1}{a_i}) = H(p_1, \ldots, p_n) + \sum_{i=1}^{n} c\log_2(a_i)$
$H(p_1, \ldots, p_n) = -c(\sum_{i=1}^{n} p_i \log_2(a_i) - \log_2(b)) = -c(\sum_{i=1}^{n} p_i \log_2(a_i) - \sum_{i=1}^{n} p_i \log_2(b)) = -c\sum_{i=1}^{n} p_i(\log_2(a_i) - \log_2(b)) = -c\sum_{i=1}^{n} p_i \log_2(\frac{a_i}{b}) = -c\sum_{i=1}^{n} p_i \log_2(p_i)$
Из непрерывности формула верна для всех $p_i \in \mathbb{R}$
Выберем $c = 1$ бит
Тогда $H = -\sum_{i=1}^{n} p_i \log_2(p_i)$ бит
Или $H = \sum_{i=1}^{n} p_i \log_2(\frac{1}{p_i})$ бит – энтропия
Флешбеш: арифметическое кодирование использует в среднем H бит на каждый символ
Отсюда арифметическое кодирование - оптимальное кодирование для данных, которые можно аппроксимировать случайным источником
Ограничение в H бит на символ называют *энтропийным барьером*
Энтропийный барьер можно преодолеть лишь учетом закономерностей

в последовательности символов

Энтропия Шеннона хорошо описывает случайные последовательности и плохо описывает "регулярные"строки (строчки, имеющие закономерности)
Для измерения информации в более сложных объектах используется *Колмогоровская сложность*
Колмогоровская сложность зависит от *декодера* и равна количеству информации, необходимому для кодирования объекта
$K_A(s) \leq K_B(s) + C_{A,B}$, где A, B – декодеры, C – константа
$K(s) \leq H(s) + C$

2 Цепи Маркова

2.1 Введение

Определение
Марковская цепь – взвешенный ориентированный граф с неотрицательными весами и суммарным весом исходящих ребер, равным 1
Пронумеруем состояния (вершины)
Пусть $b = \begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix}$ – матрица состояния B, где b_i – вероятность находиться в i-ом состоянии ($P(B = i)$)
Пусть $c = \begin{pmatrix} c_1 & c_2 & \dots & c_n \end{pmatrix}$ – матрица состояния C
Рассмотрим матрицу переходов $P = (p_{ij})_{n\times n}$, где p_{ij} – вероятность перейти из i в j
Найдем зависимость между b и c
$c_i = P(C = i) = \sum\limits_{j=1}^{n} P(C = i|B = j)P(B = j) = \sum\limits_{j=1}^{n} p_{ji}b_j$
Отсюда $c = b \cdot P$
Тогда распределение вероятностей на i-ом шаге $b^i = b^0P^i$, где b^0 – начальное состояние

Рассмотрим цепь Маркова как граф
Вершина в цепи Маркова называется *состоянием*
Поглощающее(существенное) состояние – состояние с кольцевым ребром веса 1

Цепь Маркова называется *поглощающей*, если из любого состояния можно попасть в поглощающее
Пример непоглощающей цепи: цепь с циклом длины 2 и более, где все ребра веса 1
Эргодический класс – компонента сильной связности графа Марковских цепей
Компонента сильной связности – максимальное по включению множество вершин, где из каждой можно дойти до каждой
(класс эквивалентности для отношения достижимости)
Эргодический класс называется *поглощающим*, если из него нет исходящих переходов
Цепь Маркова можно представить как граф эргодических классов (но оценить веса ребер не всегда просто)
Цепь Маркова называется *эргодической*, если она содержит ровно 1 эргодический класс

Эргодический класс называется *периодическим с циклом d*, если любая длина цикла в этом классе делится на $d > 1$. Иначе – *регулярной*

Теорема о классификации Марковских цепей
Любая Марковская цепь содержит поглощающие эргодические классы
Марковская цепь с вероятностью 1 рано или поздно оказывается в состоянии из поглощающего эргодическим классом
Для непериодического поглощающего эргодического класса в случае попадания в него существует предельное распределение вероятностей b : $b = bP$
Для любого распределения b^0 $b^0 P^n \to b$

Для цепей Маркова существуют две независимые задачи: задача поглощения – задача определения, в какой поглощающий эргодический класс мы попадем, и задача стационарного распределения внутри поглощающего эргодического класса
Займемся задачей поглощения (т.е. определим, в какой эргодический класс мы попадем)
В ходе решения этой задачи поглощающие эргодический классы можно заменить на одно поглощающее состояние
Занумеруем состояния так, чтобы сначала шли непоглощающие, а потом поглощающие

Пусть $1 \dots m$ – непоглощающие состояния, $m+1 \dots n$ – поглощающие
Тогда $P = \begin{pmatrix} Q & R \\ \mathbb{0} & I \end{pmatrix}$, где
$Q = P[1 \dots m][1 \dots m]$
$R = P[1 \dots m][m+1 \dots n]$
$\mathbb{0} = P[m+1 \dots n][1 \dots m]$ – нулевая матрица
$I = P[m+1 \dots n][m+1 \dots n]$ – единичная матрица
Возьмем матрицу состояния $b = \begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix}$
Пусть $a = \begin{pmatrix} b_1 & b_2 & \dots & b_m \end{pmatrix}$
$a^n = a^0 Q^n$
Лемма
$Q^n \to \mathbb{0}$
Доказательство
Пусть L – максимальная длина кратчайшего пути от i до поглощающей
Найдем $X = Q^L$
$x_{ij} = \sum\limits_{k_1, k_2, \dots, k_{L-1}} q_{ik_1} q_{k_1 k_2} \dots q_{k_{L-1} j}$
$\sum\limits_{j\text{– непогл.}} x_{ij} = \sum\limits_{k_1, k_2, \dots, k_{L-1}, j} q_{ik_1} q_{k_1 k_2} \dots q_{k_{L-1} j} = \delta_i < 1$ – т.к. это вероятность пройти от i до непоглощающего состояния (если бы до любого состояния, то было бы 1)
Отсюда $\max\limits_{i=1\dots m} \sum\limits_{j\text{– непогл.}} x_{ij} = \max \delta_i = \delta < 1$
Тогда $Q^n = Q^L Q^{n-L}$
Пусть $\max Q^{n-L} = v_{n-L}$
$Q^n_{ij} = (Q^L Q^{n-L})_{ij} = \sum\limits_k Q^L_{ik} Q^{n-L}_{kj} \le \sum\limits_k Q^L_{ik} v_{n-L} \le \delta v_{n-L}$
$v_n \le \delta^{\lfloor \frac{n}{L} \rfloor} \to 0$
Тогда $Q^n \to \mathbb{0}$
Теорема о поглощении
Поглощающая Марковская цепь переходит в состояние поглощения с вероятностью 1
Доказательство
Следует из леммы

Научимся определять, где же мы поглотимся
Для этого найдем мат. ожидание времени до поглощения

b^0 – начальное распределение
T – случайная величина – число шагов до поглощения
$T = \sum\limits_{i=1}^{m} T_i$, где T_i – число посещений i-ого состояния
$T_i = \sum\limits_{j=0}^{\infty} T_{ij}$, где $T_{ij} = \begin{cases} 1, & \text{если на } j\text{-ом шаге мы в состоянии } i \\ 0, & \text{иначе} \end{cases}$

Лемма

$\sum\limits_{j=0}^{\infty} Q^j = (I-Q)^{-1}$

Доказательство

$(I-Q)(I+Q+Q^2+\ldots+Q^n) = I+Q+Q^2+\ldots+Q^n-Q-Q^2-\ldots-Q^{n+1} = I-Q^{n+1} \to I$

Определение

$N = (I-Q)^{-1}$ – фундаментальная матрица поглощения Марковской цепи

$$ET = \sum_{i=1}^{m} ET_i = \sum_{i=1}^{m}\sum_{j=0}^{\infty} ET_{ij} = \sum_{i=1}^{m}\sum_{j=0}^{\infty} P(\text{цепь в состоянии } i \text{ на шаге } j) = \sum_{i=1}^{m}\sum_{j=0}^{\infty}(a^0Q^j)_i = \sum_{i=1}^{m}(\sum_{j=0}^{\infty} a^0Q^j)_i = \sum_{i=1}^{m}(a^0\sum_{j=0}^{\infty}Q^j)_i = \sum_{i=1}^{m}(a^0N)_i = a^0N\mathbb{1}$$

Заметим, что $a^0N = \begin{pmatrix} ET_1 & ET_2 & \ldots & ET_m \end{pmatrix}$

$$P(\text{погл. в } j) = \sum_{i=1}^{m} P(\text{погл. в } j \text{ из } i)P(\text{быть в } j) = \sum_{t=0}^{\infty}\sum_{i=1}^{m} P(\text{погл. в } j \text{ из } i)P(\text{быть в } i \text{ на шаге}$$

$$\sum_{t=0}^{\infty}\sum_{i=1}^{m} R_{i,j-m}P(\text{быть в } i \text{ на шаге } t) = \sum_{i=1}^{m} R_{i,j-m}\sum_{t=0}^{\infty} P(\text{быть в } i \text{ на шаге } t) = \sum_{i=1}^{m}(a^0N)_iR_{i,j-m} = (a^0NR)_{j-m}$$

Отсюда $A = \begin{pmatrix} P(\text{погл. в } m+1) & P(\text{погл. в } m+2) & \ldots & P(\text{погл. в } n) \end{pmatrix} = a^0NR$

Эргодическая теорема для регулярных цепей

Пусть Марковская цепь такова, что $\forall\, i,j\ p_{ij} > 0$ (данная цепь непериодическая)
Тогда $\exists\, b\ \forall\, b^0\ b^0P^n \to b$
(Отсюда $b = bP$, т.к. $bP = \lim\limits_n bP^{n+1} = \lim\limits_n bP^n = b$)

Доказательство
Рассмотрим $b^0 A$:
Предположим, что $\forall j \ a_{ji} = \widetilde{a}_i$
$$(b^0 \cdot A)_i = \sum_{j=1}^n b_j^0 a_{ji} = \underbrace{\sum_{j=1}^n b_j^0}_{1} \widetilde{a}_i = \widetilde{a}_i$$
Докажем, что $P^t \to A : \forall j \ a_{ji} = \widetilde{a}_i$
Пусть $m_i^n = \min_j (P^t)_{ji}$
$M_i^n = \max_j (P^t)_{ji}$
Лемма
$M_i^t - m_i^t \to 0$
Доказательство
$\delta := \min_{ij} p_{ij}, \delta > 0$(из условия теоремы)
Рассмотрим P^{t+1}:
$$p_{ji}^{t+1} = \sum_{k=1}^n p_{jk} p_{ki}^t \le \sum_{k=1, k \ne \text{posMin}}^n p_{jk} M_i^t + p_{j\,\text{posMin}} m_i^t = \underbrace{\sum_{k=1}^n p_{jk}}_{1} M_i^t + p_{j\,\text{posMin}} (m_i^t - M_i^t) \le M_i^t + \delta(m_i^t - M_i^t)$$
Аналогично $m_i^t + \delta(M_i^t - m_i^t) \le p_{ji}^{t+1}$
Отсюда $M_i^{t+1} - m_i^{t+1} \le (M_i^t - m_i^t)(1 - 2\delta) \le (1 - 2\delta)^{t+1} \to 0$

Научимся искать b
$bP = b$
Заметим, что у данном системы есть одно или бесконечно много решений
Утверждается, что $\operatorname{rg} I - P = n - 1$
Тогда пространство решений одномерное
Тогда $\exists ! b : \sum_i b = 1$

Т.о. найти b можно двумя способами:

1. $b = b^? \lim_n P^n$, где $b^?$ – любое начальное состояние

2. $b : (I - P)b = \mathbb{0}, \sum b_i = 1$ – СЛОУ

Соединим теоремы:
Пусть у нас есть Марковская цепь без периодических классов

Для начала представим, что внутри всех поглощающих классов сами состояния являются поглощающими(т.е. удалим внутренние ребра поглощающих классов и добавим петли)
Теперь мы можем определить вероятность попадания в каждое состояние каждого поглощающего класса
b^0NR – наше распределение
Теперь рассмотрим эргодический класс A
Пусть $\widetilde{p} = \sum\limits_{a \in A}(b^0NR)_a$
$\widetilde{b}^0 = \sum\limits_{a \in A}(b^0NR)_a \Big|_A \dfrac{1}{\widetilde{p}}$ – начальное состояние внутри эргодического класса A
По теореме $\exists\, b : \widetilde{b}^0 A^n \to b$
Тогда конечное распределение – объединение всех $b\widetilde{p}$

3 Формальные языки

3.1 Конечные автоматы

Пусть Σ – алфавит
$\Sigma^* = \bigcup\limits_{i=0}^{\infty} \Sigma^i$
Тогда $L \subset \Sigma^*$ – формальный язык
Пусть $\epsilon \in \Sigma^0$
Задать формальный язык можно 2 способами:

1. через порождение (генерация из существующих элементов)
2. через распознавание (через выделение элементов из множества по некоторому критерию)

Спойлер на будущее
Существуют задачи, которые вообще не решаются на компьютере
Языки делятся на 2 класса (по Хомскому)

1. Регулярные языки
2. Контекстно-свободные языки

Определение
Рассмотрим языки в алфавите $\Sigma = \{c_1, \dots, c_n\}$: $\varnothing, \{\epsilon\}, \{c_1, \}, \{c_2\}, \dots, \{c_n\}$ – *регулярные языки нулевого уровня* Reg_0
Регулярные операции над языками:

1. $L, M \mapsto L \cup M$
2. $L, M \mapsto LM = \{x : x = yz, y \in L, z \in M\}$ – конкатенация
3. $L \mapsto L^* = \bigcup\limits_{i=0}^{\infty} L^i$ – замыкание Клини
   Делает из конечного языка бесконечный
   $L^0 = \{\epsilon\}$
   $\varnothing^0 = \{\epsilon\}$

Иногда используют запись $abc := \{abc\}$ (опускают скобки)
Также $(abc)^* := \{abc\}^*$

$\mathrm{Reg}_1 = \mathrm{Reg}_0 \cup \{L \cup M : L, M \in \mathrm{Reg}_0\} \cup \{LM : L, M \in \mathrm{Reg}_0\} \cup \{L^* : L \in \mathrm{Reg}_0\}$
$\mathrm{Reg}_{i+1} = \mathrm{Reg}_i \cup \{L \cup M : L, M \in \mathrm{Reg}_i\} \cup \{LM : L, M \in \mathrm{Reg}_i\} \cup \{L^* : L \in \mathrm{Reg}_i\}$
Тогда *регулярные языки* – $\mathrm{Reg} = \bigcup\limits_{i=0}^{\infty} \mathrm{Reg}_i = \lim\limits_{i \to \infty} \mathrm{Reg}_i$
Определение
Академические регулярные выражения – выражения, задающие регулярные языки
Пусть L задается ϕ, M задается ψ
Тогда $L \cup M$ задается $(\phi)|(\psi)$ (минимальный приоритет операции)
LM – $(\phi)(\psi)$ (средний приоритет операции)
L^* – $(\phi)*$ (максимальный приоритет операции)
Определение
Конечный автомат – модель устройства, которое находится в одном из конечного количества состоянии в каждый момент времени
Модель задается:

1. Множеством состояний Q
2. Алфавитом Σ

3. Переходами $\sigma : Q \times \Sigma \to Q$

4. Начальным состоянием $S \in Q$

5. Терминальными (допускающими) состояниями $T \subset Q$ (состояниями, в которых он может находиться в конце)

$L(A) = \{x : A \text{ допускает } x\}$
Если $\exists\, A : \underbrace{L(A)}_{\text{рег. выр.}} = L$, то L – автоматный язык, $L \in \mathrm{Aut}$

3.2 Распознавание

Σ – алфавит
Q – состояние автомата
Пусть $s \in Q, T \subset Q$
$\sigma : Q \times \Sigma \to Q$ – функция перехода
$Q \times \Sigma^* = \mathrm{Conf}$
$\langle p, , \rangle \alpha \vdash \langle q, \beta \rangle$ – переход от состояния p и строки α к q, β
$c = \alpha[1]$
$\alpha = c\beta, \beta = c^{-1}\alpha$
$q = \sigma(p, c)$
$\vdash \subset \mathrm{Conf}^2$ – отношение перехода за один шаг
$\vdash^* = \bigcup\limits_{i=0}^{\infty} \vdash^i$ – существует путь
$L(A) = \{w : \langle s, w \rangle\}$
Теорема Клини
Язык регулярный $\Leftrightarrow \exists$ для него детерменированный конечный автомат
Определение
Недетерменированный конечный автомат – автомат, где может быть несколько переходов по одному символу
Недетерменированный конечный автомат *допускает* строку x, если существует путь, соответствующий x и приводящий к допуску
Тогда $\sigma : Q \times \Sigma \to 2^Q(\mathbb{P}(Q))$
$\langle p, \alpha \rangle \vdash \langle q, \beta \rangle$
$\alpha = c\beta$
$p \in \delta(p, c)$
$L(A) = \{w : \langle s, w \rangle \vdash^* \langle t, \epsilon \rangle, t \in T\}$

Теорема
Язык можно задать ДКА $\Leftrightarrow$ язык можно задать НКА
Доказательство $\Rightarrow$
ДКА – ч.с. НКА
Доказательство $\Leftarrow$
$c = |\Sigma|, 0 \dots z - 1$
$n = |Q|, 0 \dots n - 1$
s – символ
$q := \sigma$

```
    q: int[n][z] #переход
    t: set<int> #хорошие состояния
    def dfa_accept(w):
        cur = s
        for c=0...|w|-1:
            c=w[i]
            cur = q[cur][c]
        return cur in t


    q: set<int>[n][m]
    can[i][u] #можем ли мы, прочитав i символов, попасть в u
    def nfa_accept(w):
        can[0][s]=True
        for i=0...|w|-1:
            c=w[i]
            for u=0...n-1:
                if can[i][u]:
                    for v in q[u][c]:
                        can[i+1][v]=True
        return any(can|w|)
```

Пусть у нас был автомат $A_{nfa} = \langle \Sigma, Q, s, T, \sigma \rangle$
Построим $A_{dfa} = \langle \Sigma, 2^Q, \{s\}, \widetilde{T}, \widetilde{\sigma} \rangle$
$\widetilde{T} = \{M : M \cap T = \varnothing\}$
$\widetilde{\sigma} = \bigcup\limits_{u \in M} \sigma(u, c)$
Т.о. мы построили ДКА, принимающий наш алфавит
(Данный метод – конструкция подмножеств)
Но в такой конструкции многие вершины недостижимы
Поэтому вместо нее используют ленивую конструкцию (алгоритм Томпсона)

Далее добавим в автомат ϵ переходы и ϵ–НКА
ϵ-переход – переход по пустой строке (не съедает символ)
Теорема
A распознается ϵ-НКА $\Leftrightarrow$ A распознается НКА
Доказательство $\Leftarrow$
Очевидно, т.к. НКА - ч.с. ϵ–НКА
Доказательство $\Rightarrow$
Применим ϵ-замыкание

1. Добавим ϵ-ребро между p и q, если между ними есть путь из ϵ ребер
   Теперь мы не делает двух ϵ-переходов подряд

2. Сделаем терминальное состояние из тех состояний, которые соединены с терминальным ϵ переходом
   Теперь мы не делаем ϵ-переход в конце

3. Если есть переход $p \xrightarrow{\epsilon} l \xrightarrow{c} q$, добавим ребро c между p и q Теперь ϵ-переходами можно не пользоваться

4. Удалим ϵ-переходы

Теперь мы получили эквивалентный НКА
Доказательство теоремы Клини

1. Докажем $\mathrm{Reg} \subset \mathrm{Aut}$
   Рассмотрим НКА с одним начальным и одним конечным состоянием
   Построим автоматы для Reg_0 (очев)
   Будем строить по индукции
   $A \cup B$ – расположим автоматы параллельно
   AB – расположим автоматы последовательно
   A^* – Пусть p, q – начальное и конечное состояния. Построим $p \xrightarrow{\epsilon} A \xrightarrow{\epsilon} q$, соединим $p \xrightarrow{\epsilon} q, q \xrightarrow{\epsilon} p$

2. Докажем $\mathrm{Reg} \supset \mathrm{Aut}$
   Пусть $Q = \{1, \dots, n\}$
   $\xi_{i,j,k}$ – выражение, переводящее автомат из i в j, используя символы с номерами $\leq k$
   $\xi_{i,i,0} = \epsilon | c | \dots$

$\xi_{i,j,0} = c| \dots$
$\xi_{i,j,k} = \xi_{i,j,k-1}|\xi_{i,k,k-1}\xi^*_{k,k,k-1}\xi_{k,j,k-1}$
Т.о. строки, которые допускает НКА $\phi = \xi_{s_1,t_1,n}|\xi_{s_2,t_2,n}| \dots$, где $t_i \in T$, s_i – начальное состояние

Т.о. мы построили биекцию между автоматами и регулярными выражениями
Заметим, что не для всех языков можно построить конечный автомат
Докажем, что нельзя построить автомат для ПСП (правильных скобочный последовательностей)
Утверждение
ПСП – не регулярный
Доказательство
Пусть ПСП регулярный, A – ДКА для ПСП, n – число состояний A
Зададим семейства строк:

$$\begin{array}{cc} (& q_1 \\ ((& q_2 \\ (((& q_3 \\ \vdots & \vdots \\ (^{n+1} & q_{n+1} \end{array}$$

Дадим их нашему автомату
Т.к. состояний n, а строчек $n+1$, то какие-то две строчки приведут к одинаковому состоянию (по принципу Дирихле)
Пусть q_i и q_j приводят к одинаковому состоянию
Рассмотрим строчки $x = (^i)^i, y = (^j)^i$
Заметим, что автомат допускает $x \Leftrightarrow$ автомат допускает y, но x – ПСП, а y – не ПСП
Лемма о разрастании/накачке
Пусть L – регулярный
Тогда $\exists\, n > 0 : \forall\, w : w \in L, |w| \geq n\ \exists\, x, y, z : w = xyz, y \neq \epsilon, |xy| \leq n, \forall\, k \geq 0\ xy^kz \in L$
Применение для ПСП
Для фиксированного n: $w = (^n)^n, x = (^a, y = (^b, b > 0, z = (^{n-a-b})^n, k = 2, (^{n+b})^n \notin L$ – Отсюда ПСП не регулярный
Доказательство
Пусть L регулярный язык
A – ДКА для L, n – число состояний
Рассмотрим $w \in L, |w| \geq n$

Пусть при обработке строки мы прошли по следующему пути:
$\to \underbrace{u_0 \xrightarrow{w_1} u_1 \xrightarrow{w_2} \ldots \xrightarrow{w_n} u_n}_{n+1} \ldots \to t$
Тогда по принципу Дирихле $\exists\, i \neq j : u_i = u_j$
$x := w[1 : i+1], y := w[i : j+1], z := w[j+1 :]$
Тогда $xy^k z$ допустимое A
Определение
Пусть A – Д.К.А.
Состояния u, v – *различимы* строкой s, если:
$u \xrightarrow{S} x$
$v \xrightarrow{S} y$
$x \in T \oplus y \in T$ (T – терминальные состояния)
Состояния a, b эквивалентны($\sim$), если они не различимы никакой строкой s
$a \sim b$ – отношение эквивлентности
Лемма
$u \sim v \Rightarrow \sigma(u, c) \sim \sigma(v, c)$ для любого $c \in \Sigma$
(σ – переход по символу c)
Доказательство
$\sigma(u, c), \sigma(v, c)$ различимы для $S \Rightarrow u, v$ различимы cS
Алгоритм нахождения неэквивалентных состояний
Пусть D_k – множество пар состояний, различимых строкой $s : |s| \leq k$
$D_0 = \{(u, v) : u \in T \oplus v \in T\}$
$D_k = D_{k-1} \cup \{(u, v) : \exists\, c \in \Sigma : (\sigma(u, c), \sigma(v, c) \in D_{k-1})\}$
$D_k = D_{k-1} \Rightarrow D_{k+1} = D_k$
Т.к. пар конечное количество, то $\exists\, k : D_k = D_{k+1}$. Тогда далее все множества D_{k+i} будут равны D_k
Т.е. мы найдем все пары за конечное время

```
очередь Q
поместим D0 в Q, D0 в D
In - множество входящих ребер
while not Q.empty():            #n^2 раз
    (u,v) = Q.pop()
    for c in Sigma:             #|Sigma| раз
        # Тк.. всего в графе n ребер по символу c, то следующие циклы выполнятся суммарнопо( всем итерациям) n^2 раз
        for a in In[u][c]:
            for b in In[v][c]:
```

```
                    if (a,b) not in D:
                        Q.push((a,b))
                        D.add((a,b))
```

Сложность алгоритма – $O(n^2|\Sigma|)$

Теорема

Пусть A – ДКА для L, не содержащий эквивалентных состояний, и все состояния достижимы из стартового

Тогда

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

Доказательство

Рассмотрим автоматы для L – A и B

Пусть A – автомат из теоремы, B содержит меньше состояний

Возьмем автомат $A \cup B$, где между A и B нет ребер

Хотя в данном автомате 2 стартовых состояния, алгоритм поиска эквивалентных состояний будет работать корректно

Заметим, что стартовые состояния в автоматах эквивалентны

Пусть в A можно попасть в u из S_A(стартового), используя строку x

Тогда в B $v = \sigma(S_B, x)$ – эквивалентно u

Отсюда каждому состоянию из A можно сопоставить состояние из B

Т.к. в B меньше состояний, то какому-то состоянию в B эквивалентны два состояния из A, но тогда они эквивалентны между собой, что противоречит условию A

Тогда A минимально, ч.т.д.

Пусть A, B – автоматы из теоремы

По аналогичным рассуждениям существует биекция между состояниями A и B, ч.т.д.

3.3 Абстрактные штуки

Рассмотрим A – множество языков

Пусть A – хорошее, если оно замкнуто относительно регулярных операций, т.е. $R, S \in A \Rightarrow R \cup S \in A, RS \in A, R^* \in A$

Good – множество всех хороших языков

$\text{Reg}, \varnothing, 2^{\Sigma^*} \in \text{Good}$

Теперь пусть хорошие языки обязаны содержать Reg_0
Тогда $\varnothing \notin \mathrm{Good}$
Лемма
Пусть $A_\gamma, \gamma \in \Gamma, A_\gamma \in \mathrm{Good}$
Тогда $(\bigcap\limits_{\gamma\in\Gamma} A_\gamma) \in \mathrm{Good}$
Теорема
$\bigcap\limits_{A\in\mathrm{Good}} A = \mathrm{Reg}$
Доказательство
Докажем, что $\mathrm{Reg} \in \bigcap\limits_{A\in\mathrm{Good}} A$
$\forall\, i\ \forall\, A \in \mathrm{Good}\ \ \mathrm{Reg}_i \subset A$
База: $i = 0$
Переход $i \to i+1$:
$T \in \mathrm{Reg}_{i+1} \Rightarrow T = R \cup S, R, S \in \mathrm{Reg}_i \Rightarrow R \in A, S \in A \to T \in A$
$\mathrm{Reg} = \bigcup\limits_{i=0}^{m} \mathrm{Reg}_i \subset A \Rightarrow \mathrm{Reg} \subset A$
Докажем $\bigcap\limits_{A\in\mathrm{Good}} \subset \mathrm{Reg}$
$\mathrm{Reg} \in \mathrm{Good}$

3.4 Решение уравнений в регулярных выражениях

Решим линейное уравнение в регулярных выражениях
α, β – регулярные языки
X – язык
$X = \alpha X + \beta, (+) := (|)$
$\beta \subset X \Rightarrow \alpha\beta \subset X \Rightarrow \ldots \Rightarrow \alpha^*\beta \subset X$
Теорема

1. $\epsilon \notin \alpha$
   $X = \alpha^*\beta$

2. $\epsilon \in \alpha$
   $X = \alpha^*\beta | T\ \forall T$

Доказательство
Пусть $w \in X, w \notin \alpha^*\beta, |w| \to \min$

$w \in \alpha X + \beta \Rightarrow w \in \alpha X \Rightarrow w = xy, x \in \alpha, y \in X$
Если $\epsilon \notin \alpha$
$|x| > 0 \Rightarrow |y| < |w| \Rightarrow y \in \alpha^*\beta \Rightarrow w \in \alpha^*\beta$

Решим систему уравнений
$$\begin{cases} X = \alpha X + \beta Y + \gamma \\ Y = \xi X + \nu Y + \theta \end{cases}$$
$$\begin{cases} X = \alpha^*(\beta Y + \gamma) \\ Y = \xi\alpha^*\beta Y + \xi\alpha^*\gamma + \nu Y + \theta \end{cases}$$
$$\begin{cases} X = (\xi\alpha^*\beta + \nu)^*(\xi\alpha^*\gamma + \theta) \\ Y = \alpha^*\beta(\xi\alpha^*\beta + \nu)^*(\xi\alpha^*\gamma + \theta) + \alpha^*\gamma \end{cases}$$

Решим систему уравнений
$$\begin{cases} X_1 = \alpha_{11}X_1 + \alpha_{12}X_2 + \ldots + \alpha_{1n}X_n + \beta_1 \\ \vdots \end{cases}$$
$$\begin{cases} X_1 = \alpha_{11}^*(\alpha_{12}X_2 + \ldots + \alpha_{1n}X_n + \beta_1) \\ \vdots \end{cases}$$

3.5 Свойства регулярных языков

$R, S \in \mathrm{Reg}$

1. $R \cup S, RS, R^* \in \mathrm{Reg}$

2. $R \cap S \in \mathrm{Reg}$
   Доказательство
   Воспользуемся "произведением автоматов
   Пусть $Q = Q_R \times Q_s$
   $\sigma(\langle u, v\rangle, c) = \langle \sigma_R(u, c), \sigma_S(v, c)\rangle$
   $T = T_R \times T_S$

3. $\overline{R} \in \mathrm{Reg}$, где $T_{\overline{R}} = T_R^C$

4. $R_1, \ldots, R_k \in \mathrm{Reg}$
   $f : \mathbb{B}^k \to \mathbb{B}$
   $f(R_1, \ldots, R_k) = \{w | f(w_1 \in R_1, \ldots, w_k \in R_k)\}$ – регулярный

5. Пусть даны алфавиты $\Sigma, \Pi, f : \Sigma \to \Pi$
   Зададим $f^* : \Sigma^* \to \Pi^*, f^*(c_1, \ldots, c_n) = f(c_1)f(c_2)\ldots f(c_n)$ – гомоморфизм

$f^*(\alpha\beta) = f^*(\alpha)f^*(\beta)$
$f = f^*, f(L) = \{f(w), w \in L\}$
Пусть $R \in \mathrm{Reg}, f := f^*, f(L) = \{f(w)|w \in L\}, R \in \mathrm{Reg}$
Тогда $f(R) \in \mathrm{Reg}$

6. $f^{-1}(L) = \{w|f(w) \in L\}$
   Если $R \in \mathrm{Reg}, f$ – гомоморфизм
   Тогда $f^{-1}(R) \in \mathrm{Reg}$

3.6 Алгоритмический анализ регулярных языков

1. R не пуст?
   Проверяем, что терминальная вершина достижима из стартовой

2. R бесконечен?
   Ищем цикл в автомате

3. $R = S$?
   Проверяем изоморфность $A_R^{\min}$ и $A_S^{\min}$
   Или используем алгоритм поиска эквивалентных состояний

4. Количество слов длины l в языке R
   Динамика

3.7 Контекстно-свободные языки

Рассмотрим ПСП
ϵ – ПСП
A, B – ПСП $\Rightarrow AB$ – ПСП
A – ПСП $\Rightarrow (A)$ – ПСП
Рассмотрим арифметические выражения
$F \Rightarrow n$ – операция максимального приоритета (число)
$F \to -F$
$F \to (E)$
$T \to F$ – операция умножения
$T \to T * F$
$E \to T$ – операция сложения
$E \to E + T$
$E \to E - T$

Определение
Контекстно-свободная грамматика:
Σ – алфавит (терминалы)
N – множество переменных (нетерминалы)
$S \in N$ – стартовый нетерминал
P – правила
$P \subset \underbrace{N}_{\text{левая часть}} \times \underbrace{(\Sigma \cup N)^*}_{\text{правая часть}}$
Для ПСП:
$\Sigma = \{(,)\}$
$N = \{S\}$
$S = S$
Получим строчку $(())()$
$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((S))S \Rightarrow (())S \Rightarrow (())(S) \Rightarrow (())()$ – левосторонний вывод(раскрываем левый нетерминал)
$S \Rightarrow SS \Rightarrow S(S) \Rightarrow S() \Rightarrow (S)() \Rightarrow ((S))() \Rightarrow (())()$ – правосторонний вывод(раскрываем правый нетерминал)
Разбор можно изображать в виде дерева
Результат – крона дерева разбора – получается выводом всех терминалов в порядке слева направо
"Контекстно-свободные потому что правила не зависят от контекста, т.е. того, что находится вокруг

Пусть $\alpha, \beta \in (\Sigma \cup N)^*$ – сентенциальная форма предложения
$\{x | S \Rightarrow^* x, x \in \Sigma^*\}$ – язык грамматики

Г – К.С.Г.
Язык – К.С., если существует К.С.Г. Г
Регулярные языки $\subset$ К.С. языки
Определение
Г – праволинейная грамматика, если ее правила имеют следующий вид:
$A \to c$
$A \to cB$, где A, B – нетерминалы, c – терминал
Обозначения
$a, b, c, \ldots \in \Sigma$
$\ldots, u, v, w, x, y, z \in \Sigma^*$
$\alpha, \beta, \beta, \xi, \ldots \in (N \cup \Sigma)^*$

$A, B, C, \ldots \in N$
Теорема
L – регулярный $\Leftrightarrow$ L задается праволинейной КСГ
Доказательство $\Rightarrow$
Рассмотрим автомат
Сделаем нетерминальные вершины нетерминалами
Если есть переход $A \xrightarrow{c} B$, добавим правило $A \to cB$
Если A терминальная, то добавим правило $A \to \epsilon$
S – стартовое состояний
Доказательство $\Leftarrow$
Избавимся от правил $A \to c$:
$A \to cY, Y \to \epsilon$
Теперь выполним обратное преобразование и получим автомат

Т.к. грамматик счетное множество, а языков – несчетное, то не каждый язык можно задать грамматикой

$0^n 1^n 2^n$ и $\alpha\alpha$ – не контекстно-свободные
Заметим, что у языка может быть несколько деревьев разбора
Определение
Грамматика называется однозначной, если у любого слова не более одного дерева разбора
Грамматика называется существуенно неоднозначной, если она не имеет однозначной к.с. грамматики

A, B – к.с.
$A \cup B, AB, A^*$ – к.с.
Γ_A, Γ_B
$A^* : S \to \epsilon, \Sigma \to AS$
$A \cup B : S \to A, S \to B$
$AB : S \to AB$

$A \cap B$ – необязательно к.с.
Пример: $0^n 1^n 2^m, 0^m 1^n 2^n$ – к.с., но $0^n 1^n 2^m \cap 0^m 1^n 2^n = 0^n 1^n 2^n$ – не к.с.

3.8 Преобразование грамматик. Нормальная форма Хомского

Определение

Нормальная форма Хомского – грамматика, все правила которого имеют следующий вид:
$A \to BC$
$A \to c$
$S \to \epsilon$, но если это правило есть, S не встречается в правых частях правил
Символ грамматики называется бесполезным, если он не может встретиться в корректном выводе
Символ бесполезен в следующих двух случаях:

1. он недостижимый
2. он непорождающий (т.е. он не приводит к корректному выводу)

Теорема

Если нет недостижимых и порождающих, то все символы полезные
Сначала нужно избавиться от непорождающих, а затем от недостижимых (т.к. удаление непорождающих может приводить к появлению недостижимых)

Научимся приводить грамматику в НФХ
Все символы c в правилах, содержащих справа нетерминалы, заменим на X_c и добавим правило $X_c \to c$
Некоторые правила примут вид $A \to B_1 B_2 B_3 \ldots B_n$
Заменим такое правило на $A \to B_1 B_{2\ldots n}, B_{2\ldots n} \to B_2 B_{3\ldots n}, \ldots, B_{n-1\ldots n} = B_{n-1} B_n$
Теперь избавимся от ε

```
while change:
    for A -> a:
        if a consists of e-gen:
            mark A as e-gen
```

Мы нашли все ε-порождающие нетерминалы
Рассмотрим правило $A \to BC$
Если B – ε-порождающий: Добавим правило $A \to C$

Если C – ε-порождающий: Добавим правило $A \to B$
Теперь мы можем удалить все правила $A \to \varepsilon$
Осталось обработать случай $S \to \varepsilon$, где S – стартовый
Если S – ε-порождающий:
S' – новый старт
$S' \to S$
$S' \to \varepsilon$
Осталось избавиться от цепных правил: $A \to B$
Заметим, что данные правила похожи на ε-переходы в автоматах
Выполним аналогичные действия для избавления от них
Сначала сделаем транзитивное замыкание:
Для правил $B_1 \to B_2, B_2 \to B_3, \ldots, B_{n-1} \to B_n$ добавим правила $B_i \to B_j, i < j$
Теперь в разборе можно избежать двух цепных правил подряд
Для правил $A \to B, B \to CD$ добавим правило $A \to CD$
Теперь цепными правилами можно не пользоваться. Удалим их
Заметим, что действия надо выполнять в данном порядке
Алгоритм Кока-Янгера-Касами (для разбора грамматики в НФХ)
Дана КС Γ в НФХ и слово ω
Нужно проверить, принадлежит ли слово грамматике
$dp_A[l][r] = \begin{cases} 1, & A \Rightarrow^* \omega[l,r) \\ 0, & \text{иначе} \end{cases}$
Если в грамматике есть правило $A \to c, \omega[i] = c$, то $dp_A[i][i+1] = 1$
$dp_A[l][r] = \bigvee\limits_{A \to BC} \bigvee\limits_{k=l+1}^{r-1} dp_B[l][k] \wedge dp_C[k][r]$
Вычисления нужно проводить по возрастанию $r - l$
Ответ в $dp_S[0][n]$
$|\Gamma|$ – размер грамматики (количество правил)
Сложность алгоритма $O(|\Gamma| n^3)$

3.9 Автомат с магазинной памятью (стек-машина)

Ребра автомата имеют вид <что на входе>,<что берем со стека>/<что кладем в стек>
Z_0 – маркер дна, лежащий на дне стека
Математическая модель МП-автомат
Σ – входной алфавит

Π – стековый алфавит
$Z_0 \in \Pi \setminus \Sigma$ – маркер дна
Q – состояния
$s \in Q$ – стартовое состояние
$T \subset Q$ – терминальные состояния
$\delta : Q \times (\Sigma \cup \epsilon) \times \Pi \to \mathbb{P}(Q \times \Pi^*)$
$\langle q, x, \alpha \rangle \vdash \langle \gamma, y, \beta \rangle$
$x = cy$
$\alpha = A\gamma$
$\beta = \xi\gamma$
$\langle \gamma, \xi \rangle \in \delta(q, c, A)$
Теорема
КС $\Leftrightarrow \exists$ МП-автомат