

Лекция 1

1. Теория вычислительных процессов

- БЦ ВМ (бортовая цифровая вычислительная машина)
- Прямое тестирование - подать входные данные, сравнить выходные данные с ожидаемыми.

В алгоритме должны доказать и обеспечить, что решение существует, является наиболее оптимальным в рамках определенной задачи.

- Семиотика - наука о знаниях.
- Синергетика - объединение (синтез) объектов, приводящее к появлению новых свойств, которые не присущи каждому объекту по отдельности.

1.1 Цикл информации

- 1) Измерительная информация
 - 2) Первичная обработка информации (регистрация)
 - 3) Вторичная (семантическая) обработка информации (анализ)
 - 4) Формирование управляющих воздействий (должно быть **проактивным** или предиктивным)
- Таблица измененных значений - набор данных после первичной обработки.

Семиотика

- 1) Синтаксис -> первичная обработка
- 2) Семантика -> вторичная обработка
- 3) Прагматика -> формирование программы (управляющих воздействий)

Лекция 2

Алфорова З.В. - теория алгоритмов: учебное пособие для вузов
Москва, статистика 1973 год 646 с

Кузнецов О.Д, Аднецов-Вельский Т.М. Дискретная математика
для инженера. Москва. Жнергитик 1980 342 с

Чень 4 л и р, математическая логика, какое-то доказательство
теоремы

П Кравец В.Г. Любинский В.Е. - основы управления космическим аппаратом. Москва, машиностроение, 224 с Креницкий Н.А. - алгоритмы вокруг нас. Москва 1984 г, 212 с Колмогоров теория информации, теория алгоритмов 1987 г - 304с Мальцев А.И. - алгоритмы и рекурсивные функции, Москва 1965, 1986-1997 Успенский В.А, Семенов - теория алгоритмов; основные открытия и приложения, Москва 1987 год, 280+с Яблоснский С.В. - введение в дискретную математику

2. Элементы теории алгоритмов и формальных системных

2.1 Понятие алгоритма и его основные свойства

2.1.1 Определение алгоритма Алгоритм - точное предписание о выполнении в определенном порядке некоторой системы операций для решения всех задач некоторого данного типа.

аль-Харезим Абу Абдула Мухаммед бен Муса - (787 Хива - 850)

аль джебр

$ax^2 + bx + c = 0$ Р = 0 Диафант Натисевич

Главным внутри математическим применением теория алгоритмов явились: - доказательства невозможности алгоритмического решения различных математических проблем - с точки зрения практической вычислительной математики или кибернетики, алгоритм - это программа *“критерием алгоритмичности”* является возможность запрограммировать эту задачу - кроме того с точки зрения практической, нужно уметь сравнивать разные алгоритмы для решения одной и той же задачи (по качеству решения, по характеристикам самих алгоритмов, точность)

2.1.2 Основные подходы к построению алгоритмов теория алгоритмов: 1) теория рекурсивных функций - основана на понятие числовой функции и ее вычислении 2) Теория интерпретирующих систем (автоматы и машины) - алгоритм реализуется некоторым абстрактным, детерминированным устройством (машин Тьюринга) 3) Алгоритм алфавитных преобразований - основа операторы постановки и вывода на множестве слов (*формальные системы на основе формальных грамматик*) Все выше перечисленные уточнения/подходы реализации “Алгоритм” задаются с помощью средств алгоритмической системы

Под алгоритмической системой будет понимать всякий общий способ формальной задачи алгоритма

2.2 Элементы теории рекурсивных функций

2.2.1 Основные элементы примитивной-рекурсивной функции В данной функции функция не как функция отображения

Функцию для которой существует вычисляющий ее алгоритм, будем называть вычислимой функцией

рекурсия (лат *recurso*) - бегу назад

такой способ задания, когда каждое ее значение задается через значение этой же функции или какой-нибудь другой вычислимой функции для меньших и для ранее неопределенных значений аргументов. Функции задаваемые таким образом будем называть рекурсивными.

строятся на основе: - функции строятся на основе элементарных или базисных - операций или операторов формируемых на ... 1)

$0^n(x_1 \dots, x_n) = 0;$

2) функция проекции/тождества, повторяющая значения своих аргументов

$I^n(x_1, x_2, \dots, x_n) = x_m$ 3) функция непосредственно следовая
 $S(x) = x + 1$

замечания: предполагается, что все элементарные функции (базисные) являются вычислимыми

Лекция 3

Основные свойства алгоритмов (к прошлой лекции)

- 1) Детерминированность Когда одному и тому набору входных данных ставится в соответствии (вычисляется) единственный выходной набор
- 2) Массовость Алгоритм применим к некоторому множеству входных данных
- 3) Результативность Когда обеспечивается получения результата или выходного набора данных за конечное число шагов

2.2.2 Рекурсивные функции Вычислимые функции базируются на элементарных функциях: - 0-функции, - непосредственного следования, - функции проекции

Оператор суперпозиции функции - заключается в подстановке одних арифметических функций вместо других

Пусть задана m - местная функция $h_n^m(x_1, \dots, x_n)$ m - n местная $g_1^n(x_1, \dots, x_n)$ g_2^n - до g_n^n $Fm^{n=(h m_1 * g_1^n * \dots * g_m^n)} = h^m$

$F_m^n(h^m, g_1^n, \dots, g_m^n) = h^m(g_1^n(x_1, \dots, x_n), \dots, g_m^n(x_1, \dots, x_n))$ Элементарные функции и суперпозиции позволяют реализовать всевозможные функции подстановки, переименования и отождествления переменных $f(x_1, x_2, x_3, \dots, x_n) = f(I_2^2(x_1, x_2), I_1^2(x_1, x_2), x_3, \dots, x_n)$
Дано: $O^1(x) = 0$; $S^1(x) = x + 1$;

$$F_1^1(S^1, O^1) = S^1(O^1(x)) = 0 + 1 = 1;$$

Оператор примитивной рекурсии - позволяет строить $n+1$ местную функцию $F^{(n+1)}$ по 2-м заданным функциям:

$$g^n(x_1, \dots, x_n) \quad h^{n+2}(x_1, \dots, x_{n+2})$$

$$f(x, y+1) = h^{n+2}(x, y, f(x, y))$$

Д/з: построить ϕ -ию, (0-3 для x и y)

Определение Примитивно рекурсивная функция - функцию полученная из базисной элементарной функции, проекции и ... с помощью конечного числа применения оператора суперпозиции и примитивной рекурсии

Описание любой арифметической функции как примитивно-рекурсивной задает точное математическое описание алгоритма ее вычисления арифметической

Д/з: Доказать, что функция $f(x) = 2x$ примитивно-рекурсивная [[Pasted image 20240923152513.png]] ####
 $f(x_1, x_2, x_3) = x_1, x_2, x_3$ - доказать, что примитивно-рекурсивная [[Pasted image 20240923152536.png]]

2.2.3 Обще рекурсивные и частично рекурсивные функции Оператор минимизации или наименьшего корня (μ) - позволяет определить новую арифметическую функцию $f^{(n)}(x_1, \dots, x_n)$ с помощью ранее известной или построенной функции $g^{(n+1)}(x_1, \dots, x_{n+1})$: для любого заданного набора значений переменных $\langle x_1, \dots, x_n \rangle = \langle a_1, \dots, a_n \rangle$ в качестве $f...$

принимается значение наименьшего целого неотрицательного корня $y=a$ в уравнении: $* g^{(n+1)}(a_1, \dots, a_n, y) = 0$;

Замечание: Будучи примененной вычислимой функции g снова дает вычислимую функцию. Т.е. мю-оператор сохраняет свойства вычислимости. Процесс же вычислений реализуется следующим образом: сюда фотку

Пока значение g не ставит равное 0, мы не найдем корень $*$. То минимальное значение y на наборе x_1, x_n при котором $*$ является тождеством и будет значением функции $f(a_1, \dots, a_n) = y$

Однако в отличии от оператора супер позиции ... такой процесс вычислений может не привести к конкретному результату

Такое событие произойдет тогда, когда уравнение $*$ не имеет решение (корней), это может произойти в 3 различных случаях: - когда в самом начале неопределенно значение g и 0; - когда для $y = 0, 1, 2, \dots, k$ - значения функции g определены, но для уравнения g является тождеством, но на каждом шаге значение функции g не определено - когда функция $g(x_1, \dots, x_n, y)$ является всюду определенной, но ни на одном наборе своих аргументов не выполняется тождество $*$

Содержательный смысл мю-оператора Необходимо отметить, что в примитивной рекурсии отличается от суперпозиции тем, что она учитывает возможность индуктивного перехода при определении функции, а значение y - $y+1$. при этом заранее задано или известно количество цикла или повторений при вычислении значений определяемой функции. Применение же мю-оператора хот и сводится к повторному или циклическому вычислению значения определяемой функции, однако все это происходит *без наличия какой-либо априорной информации* о количестве таких повторений

Арифметические функции, которые могут быть построены из элементарных функций конечного применения оператора суперпозиции, примитивной рекурсии и операторов наименьшего корня (мю) называется частично рекурсивной функцией.

Всюду определенные частично рекурсивные функции называются общей рекурсивной функцией

Лекция 4

2.3 Значение рекурсивных функций. Тезис Черча

ЧРФ представляет собой наибольший общий класс конструктивного задания арифметических функций. Теор. алгоритм. ставится понятию конструктивизма и изучению конструктивного вычисления и алгоритмов.

Конечная базовая система функций.

Конечная система преобразований этих или других операций, которые задают большое многообразие различных алгоритмов вычисления каких-либо функций. **Тезис Черча** Класс алгоритмов или машинно вычислимых частично числовых функций совпадает с классом всех ЧРФ.

Машина Тьюринга - это тоже алгоритмическая система; абстрактный автомат.

2.3.1 Основные понятия машины Тьюринга Для задания м.Тьюринга необходимо задать : 1) Управляющее устройство, которое может находиться в одном из состояний, образующих конечное множество $Q = q_1, q_2, \dots, q_n$. 2) **Бесконечную ленту**, разбитую на ячейки, в каждой из которых может быть записано значение бесконечного алфавита A . 3) Устройство обращений к ленте, которое в каждый момент времени обозревает только одну ячейку этой ленты. 4) Набор команд

2.3.2 Принципы функционирования машины Тьюринга

В каждом такте МТ находится в одном из своих состояний множества Q . А головка обозревает одну ячейку.

В следующем такте МТ переходит в другое состояние или остается в том же.

В обозреваемую ячейку записывает символ алфавита A или остается в том же состоянии. Головка передвигается в L , R или E . [[Pasted image 20250104141841.png]] Таким образом, данные для МТ - это слова, записанные на ленте.

Каждый такт МТ заключается в считывании символа, определении состояния; в зависимости от этого, переход в новое состояние, запись в ячейку нового символа и перемещение головки.

Формально, $\{q_i a_j \rightarrow q'_i, a', \alpha_k\} = P$, где q_i и q - состояние МТ (Q) a_j , a' - обозреваемые символы на ленте до и после выполнения этой команды. (A). $\alpha_k \in L, R, E$ Совокупность команд МТ образуют программу (P)

q_0 - начальное состояние q_z - конечное состояние λ - очищение этой ячейки

Определение Полным состоянием МТ называется такая совокупность символов a и q , которая однозначно определяет дальнейшее поведение МТ.

$K_j = \alpha_1 q_i \alpha_{2i}$, где α_1 - слово на ленте, находящееся слева от обозреваемой ячейки q_i - текущее состояние α_2 - слово, образующееся символом, обозреваемым головкой и всеми символами справа от него.

Пример: $K_5 = baq_3de$

λ	b	a	d	e	λ
\uparrow					
q_3					

Стандартной начальной конфигурацией называется конфигурация вида $k_0 = q_0 \alpha$; Стандартной конечной конфигурацией называется конфигурация вида $k_z = \alpha_1 q_2 \alpha_2$

МТ будем называть абстрактный автомат, задаваемый кортежем, где Q - внутренний алфавит, A - внешний алфавит, K_0 - стандартное состояние, P - программа

Лекция 5

2.3.2 Способы задания МТ Задать вычислительный алгоритм в алгоритмической системе Тьюринга - это значит задать пример .. по любым внешними или входным данным

Поведение или функционирование МТ может быть задано программой, составленной 3 различными способами: 1) совокупностью команд (перечислением множества P); 2) таблица переходов МТ 3) диаграммой (блок-схемой) переходов МТ

Пример: 1) Совокупность команд $M = \langle Q, A, K_0, P \rangle$ $Q = \{q_0, q_1, q_2, \dots, q_z\}$ $A = \{a, b, c, \lambda\}$ $K_0 = \{q_0 abc \lambda\}$

1. $P = \{q_0 a \rightarrow q_1 \lambda R \quad q_1 b \rightarrow q_2 \lambda R \quad q_2 c \rightarrow q_z \lambda E\}$ *Очистка*

2) Таблица переходов

q/A	a	b	c	λ
q_0	$q_1 \lambda R$			

q/A	a	b	c	λ
q_1		$q_2 \lambda R$		
q_2			$q_z \lambda E$	
q_z				

- 3) Диаграмма переходов $a \rightarrow \lambda R \ b \rightarrow \lambda R \ c \rightarrow \lambda R \ (q_0) \Rightarrow (q_1) \Rightarrow (q_2) \Rightarrow (q_z)$

2.3.3 Вычисления на Машине Тьюринга В соответствии с конфигурацией МТ со всякой не заключительной применимо некоторая команда МТ, после такого применения МТ переходит в новую конфигурацию $k(i+1)$.

Если для некоторых K_i и K_j при ее функционировании \exists такая последовательность смены конфигурации из $K_i \rightarrow K_{i+1}, \dots, K_j$, то этот факт будем обозначать ..

Пусть f - некоторая функция, отображающая множество слов из алфавита Аисх в Арез в общем виде. Говорят, что МТ М правильно вычисляет функцию f , если выполняется 2 условия:
1) $f(V) = W; q_1 V \Rightarrow q_z W$ - в тетради еще штучка

- 2) . существует V ; $f(V)$ - не определено, то МТ, запущенная в стандартной начальной конфигурации $q_0 V$, работает бесконечно.

Если для некоторой функции f существует МТ М, которая ее правильно вычисляет, то f называется (правильно) вычислимой по Тьюрингу. Замечания: 2 МТ являются эквивалентными если они правильно вычисляют одну и ту же функцию

2.3.4 Тезис Тьюринга Всякий алгоритм может быть реализован на МТ.

Замечания Сформулированный тезис позволяет утверждать, что если не возможно создать Мт для вычисления какой-либо функции, то алгоритма вычисления такой функции не существует вообще

Доказательства этого тезиса не существует, поскольку само понятие алгоритма для МТ, определено неявно (интуитивно)

Подтверждением тезиса Тьюринга могут служить следующие неформальные обстоятельства: - практика создания программ для МТ по всем известным вычислимым функциям; - сводимость описания алгоритма в любой алгоритмической системе к МТ

Отсутствие алгоритма относится к решению “проблемы остановки” для произвольной МТ. Проблема определения выводимости в любой логической теории

2.4 Алгоритмическая система Поста

Эмиль Пост 1897

2.4.1 Понятие формальной системы Теория формальных систем рассматривает только синтаксические свойства изучаемых объектов - слов, то есть математические объекты в формальных системах понимаются как последовательности символов или слова в некотором фиксированном алфавите, а операции над этими объектами или словами являются операциями над символами или словами.

- 1) Формальная система называется математический кортеж $\langle A, A_1, R \rangle$, где A - конечный алфавит $A = \{a, b\}$; $|A| = 2$; $A^* = \{\emptyset, a, b, ab, ba, \dots\}$; $|A^*| = \infty$ A_1 - множество правильно построенных выражений в A (аксиом) $A_1 \in A^*$ R - количество множеств вычислимых выражений вида $R = \{(\alpha_1, \dots, \alpha_n, \beta)\} \alpha_n, \beta \in A^*$

МТ = FS - ? $M = \langle Q, A, K_0, P \rangle$ FS = $\langle A, A_1, R \rangle$ A - алфавит целых чисел A_1 - исходные данные

2.4.2 Система Туэ Аксель ДЗ: $1^4 + 1^5 = 1^{4+5}$ $1111 + 11111 =$

- 1) Совокупность команд $q_0 1 \rightarrow q_0 1R$; $q_0 + \rightarrow q_1 1R$; $q_1 1 \rightarrow q_1 1R$; $q_1 \lambda \rightarrow q_2 \lambda L$; $q_2 1 \rightarrow q_2 \lambda E$
- 2) Таблица состояний

q/A	1	+	λ
q_0	$q_0 1R$	$q_1 1R$	
q_1	$q_1 1R$		$q_2 \lambda L$
q_2	$q_2 \lambda E$		

- 3) Диаграмма переходов [[Pasted image 20250105104128.png]]

Лекция 6

2.4.2 Частный случай Туэ Аксель Определение: Система подстановок или полусистема Туэ определяется как формальная

система через следующие элементы: A (алфавит), конечное множество подстановок вида $R_i(\alpha_i, p_i) = \alpha_i \rightarrow p_i$, $\alpha_i, p_i \in A$

$\alpha_i \rightarrow p_i$ а, б интерпретируются как некоторое правило вывода как R_i

Из некоторого слова γ выводимо слово δ , для всех $\gamma, \delta \in A^*$

если δ получается из γ путем подстановки слова β_i вместо какого-либо вхождения слова α_i в это слово γ

Тогда выводом полусистемы называется цепочка $\beta\gamma\alpha : \alpha + E_1 + E_2 + \dots + E_n + \beta; \alpha \Rightarrow \beta$;

Ассоциативным исчислением или системой Туэ называется такая формальная система, определяемая некоторым алфавитом A и конечным множеством соотношения вида A_i : $a_i \leftrightarrow b_i$, каждая из которых (\leftrightarrow $a_i \rightarrow b_i$; $b_i \rightarrow a_i$) Если из a выводимо b в асс. исч, то $a + b \Rightarrow b + a$ $a \Rightarrow b \Rightarrow b \Rightarrow a$

В асс. исч отношение (\leftrightarrow) в соответствующей подстановке является отношением эквивалентности, т.е. этому отношению экв. соответствуют след. свойства: - рефлексивности $\forall(\alpha \in A^*)(\alpha \leftrightarrow \alpha)$; - транзитивности $\forall(\alpha, \beta \in A^*)(\alpha \leftrightarrow \beta \ \& \ \beta \leftrightarrow \gamma) \Rightarrow (\alpha \leftrightarrow \gamma)$ - симметричность $\forall(\alpha, \beta \in A^*)(\alpha \leftrightarrow \beta) \Rightarrow (\beta \leftrightarrow \alpha)$ Примечание: Применение асс. исч: Поскольку выводимость в АИ - отношение эквивалентности, то задачей обработки построения оптимальных планов может служить следующие определения эквивалентных отношений и определения некоторого плана

Пример:

2.4.3 Частный случай формальной системы. Определение

канонической система Поста Система Поста определяется как формальная система $FS_p < A, X, A_1, R >$, где $A = \{a_1, \dots, a_n\}$ - собственный алфавит $X = (x_1, \dots, x_n \in A^*)$ - алфавит или множество переменных A_1 - конечное множество аксиом $\in A, X$ R - конечное множество правил вывода $\gamma_1, \gamma_2, \dots, \delta$

Формальная система Поста Некоторое слово α получается применением некоторой аксиомы ω из множества в том случае, если переменных в ω подставляются слова из A^*

Некоторое слова α непосредственно выводимо из слов $\alpha_1, \dots, \alpha_n$ применением продукции из R_i с n посылками, если найдется такая подстановка слов вместо переменных в R_i , которая посылки в R_i превращает в соответственно слова $\alpha_1, \dots, \alpha_n$, а заключение правило R_i непосредственно в слове α .

Пример в тетради $acab, cabb \vdash bb$; ? $R_I = ax_1b, x_1bx_2 \vdash bx_2$
 $x_1 = ca; x_2 = b$

O2

Последовательность слов называется док-вом в система Поста, если каждое слово этой последовательности: - Либо результат применения аксиомы; - Либо непосредственно выводимо из предыдущих слов последовательности применением некоторой продукции

Слово α называется доказуемым или теоремой, если оно является последним словом некоторого доказательства

Пример: построить ФСП, перечислить все нечетные числа $A = \{1\}$ $X = \{x\}$ $A_1 = \{1\}$ $R = \{x \rightarrow 11x\}$ Если четные, то меняем аксиому $A_1 = \{11\}$ 1, 111, 11111

2.5 Алгоритмические формальные системы и проблемы разрешимости

2.5.1 Неклассические алгоритмические системы С широким распространением ЭВМ стали возникать другие виды алгоритмических системы более ориентированные к алгоритмизации задач, решаемых на компьютере. Среди них выделяют: - операторные алгоритмы Ван Хао; $i: |w| \alpha | \beta |$, i - номер приказа, w - элементарная операция над объектом, α, β - номера дальнейших приказов - операторные алгоритмы А.А. Ляпунова; $p(x < y)$, если истинно, то выполняется, если нет, то не выполняется - Блок-схемный метод алгоритмизации; ВНД - время нормированного документа, ЗПД - заработная плата документа, $ВНД = \sum_{i=1}^n k_i x t_i$ $ЗПД = \sum_{i=1}^n K_i x p_i$ - Логические схемы Ю.И. Янов

2.5.2 Разрешимые и перечислимые множества (Лекция 7)

Лекция 7

2.5.2 Разрешимые и перечислимые множества Прикладные результаты теории вычислительных процессов в основном заключаются в определении принципиальной разрешимости различных задач, требующих своего решения

Некоторое множество M называется разрешимым (или рекурсивным), если существует некоторый алгоритм A_n по любому объекту a дает ответ принадлежит ли множеству M . В этом случае

алгоритм называется разрешимой этого множества M . Множество M разрешимо, если оно обладает обще рекурсивной характеристической функцией (всюду определенной), которая дает значение 1. Множество M называется перечислимым или рекурсивно перечислимым, если это множество является областью значений некоторой общерекурсивной функции (ОРФ), т.е. существует ОРФ $\exists \psi_m(x) : \text{если } \alpha \in M \Leftrightarrow \alpha = \psi_m(x)$

Функция ψ_m называется перечисляющей для множества M , и соответственно алгоритм, вычисляющий значение этой функции является тоже перечисляющим или порождающим

Замечание (важность понятия различимости и перечисления) Благодаря применению или использованию становится точным математически “конструктивный способ задания M ” “множество M , заданное конструктивно”

Замечание С теоретической точки зрения неразрешимости какой-либо проблемы или задачи не является неудачей, а научный факт. Знание основных неразрешимостей теории алгоритмов является таким элементом научной культуры, как для физики знания невозможности создания вечного двигателя

Рекомендации: 1) Отсутствие общего алгоритма, решающего данную проблему не означает отсутствие разрешимости в частных случаях. Из этого следует - необходимо искать не разрешимые частные случаи 2) Появления неразрешимостей - это, как правило, результат чрезмерной общности постановки задач, а это означает, что задача в более общей постановке имеет больше шансов оказаться не разрешимой.

3 Автоматные модели

3.1 Основные понятия теории конечных автоматов

3.1.1 Определение Конечным автоматом называется кортеж S (пятерка) $\langle X, Q, U, \delta, \lambda \rangle$, где $X = \{x_1, \dots, x_n\}$ - входной алфавит Q - внутренний алфавит $U = \{u_1, \dots, u_n\}$ - выходной алфавит $\delta: X * Q \rightarrow Q$ - функция перехода $\lambda: X * Q \rightarrow U$ - функция выхода

$a b \rightarrow S \rightarrow a \in X * b \in U$

Функционирование конечно автомата Функционирует дискретно по тактам и шагам. Рассмотрим такт функционирования i $x(i) \in X; q(i) \in Q \rightarrow q(i+1); u(i) \in U;$

а) КНА 1-го рода (Мили) $q(i) = \delta(x(i), q(i-1)); U(i) = \lambda(x(i), q(i-1));$

б) КНА 2-го рода (Мура) $q(i) = \delta(x(i), q(i-1))$; $u(i) = \lambda(q(i)) = \lambda(\delta(x(i), q(i-1)))$

КНА S , функционирование которого всегда начинается с некоторого начального состояния $q_0 \in Q$, называется инициальным

Замечание: Обычно рассматривают конечные автоматы, в которых рассматривают $Q_p \in Q$ (множество конечных состояний) $q_i \in Q_F$ КНА останавливается (завершает свою работу)

3.1.2 Способы задания конечных автоматов

1) Создание матрицы переходов δ и λ $\triangle(\delta) \triangle(\lambda) \triangle/\triangle =$

Q/X	X1	X2
q1		
q2		q1
...		

2) Граф или диаграмма переходов

$\Gamma = \langle Q, (\delta, \lambda) \rangle$;

$x_k/u_m(q_i) \rightarrow (q_j)$

3) $M = ||M_{ij}||$ $m_{ij} = x_i/u$

q / q	q1	q2	q3
q1			
q2			x/u
q3			

Пример: $S = \langle X, Q, U, \delta, \lambda \rangle$; $X = \{a, b, c, d\}$; $Q = \{q_1, q_2, q_3\}$; $U = \{0, 1\}$

$\triangle/\triangle =$

X / Q	a	b	c	d
q_1	$q_2/0$	$q_3/0$	$q_3/1$	$q_1/1$
q_2	$q_3/0$	$q_3/1$	$q_1/1$	$q_2/0$
q_3	$q_3/1$	$q_1/1$	$q_2/0$	$q_1/0$

/ : (1)

Q/Q	q_1	q_2	q_3
q_1	$\frac{d}{1}$	$\frac{a}{0}$	$\frac{c}{1} \frac{b}{0}$
q_2	$\frac{c}{1}$	$\frac{d}{0}$	$\frac{b}{1} \frac{a}{0}$
q_3	$\frac{b}{1} \frac{d}{0}$	$\frac{c}{0}$	$\frac{a}{1}$

[[Pasted image 20250105134159.png]] $q_1 \in Q_i \subset Q_{0i} \alpha = bbcd \Rightarrow \beta = ? \beta = 0110$

$L = \{\alpha\} L = \{a^m b^n \mid m, n > 0\}$ S - ? (построить), при $U = \{0, 1\}$ $Q = ?$, если $q_i \in Q_f \Rightarrow U(i) = 0$, иначе $U(i) = 1$.

Лекция 8

3.1.3 Полнота и детерминированность Полнота: Полным КНА называют такой КНА, который имеет всюду определенную функцию перехода и выхода. $\forall x_i \in X, \forall q_j \in Q \exists q_k = \delta(q_j, x_i); U_k = \lambda(q_j, x_i);$

Частичный КНА - такой автомат, у которого $\delta \lambda$ определены не для всех пар входного алфавита X и алфавита внутреннего состояния Q.

Замечание: Для ЧКНА множество слов недопустимого для этого автомата называют множеством запрещенных слов.

Детерминированным КНА называются такой автомат S, который имеет однозначную функцию перехода δ , т.е. в графе переходов из каждой вершины должна выходить не более, чем одна дуга, помеченная одним и тем же символом.

В автоматной матрице M любой входной символ встречается в строке матрицы не более 1 раза. В противном случае - автомат недетерминирован. $b(q_i) \preceq (q_j) a$

3.2 Синтез КНА

Синтез - построение КНА по описанию множеств слов во входном алфавите, который допускаться этим КНА

Анализ - процесс обратный к синтезу - получение множества входных слов, допустимых этим КНА по известному или заданному КНА. Анализ предполагает о предъявлении слову определение допуска слова в заданному КНА.

Замечание: Синтез - процесс модели функционирования
Анализ - процесс распознавания входных слов по результатам
объекта контроля.

3.2.1 Регулярные выражения $R \sim X = \{x_1, \dots, x_n\}$ определяется рекурсивно Следующим образом: 1) $R = \emptyset$ 2) $a \in X \Rightarrow R = a$ (один символ) 3) $R_1, R_2 \wedge, \vee, () <>$, то а) $R = (R_1 \vee R_2)$; б) $R = (R_1 \wedge R_2)$; в) $R = (R_1); \bar{R} = (R_2)$

$(R)^* = \bigvee_{i=0}^{\infty} R^i = \emptyset + R + RR + RRR + \dots$ 4) Ни что другое не является регулярным выражением

Замечание: Рекурсивное определение означает такой способ задания формулы, функции, при котором эта конструкция на некотором шаге определяется через такую же конструкцию с предыдущего шага. **Пример:** $X = \{a, b, c, d\}$; 1) $R = (a \vee b) c \Leftrightarrow L(x) = \{ac, bc\}$ 2) $R = (a b c \vee a) (a b d) \Leftrightarrow L(R) = \{abcbcd, aabd\}$ 3) $R = (a^* \vee b^*) \Leftrightarrow L(R) = \{\emptyset, a, b, aa, abb, \dots\}$. 4) $(a \vee b)^* c \Leftrightarrow L(R) = \{ac, bc, aabac, bbcb, c, \dots\}$ 5) Про спутник: $L(R) = \{abcd, abcdabcbcd, \dots\}$ ДЗ: 6) $R = a^* b^* \Leftrightarrow L(R) = \{\emptyset, a, b, aa, bb, ab, \dots\}$ 7) $L(R) = \{a^n, b^m \mid n, m \geq 0\}$; $R = ?$ (см п.3) 8) $L(R) = \{a^n, b^n \mid n \geq 0\}$; $R = ?$

3.2.2 Разметка регулярных выражений $\exists R: x = \{x_1, \dots, x_n\}$ разделяющее местами, будем называть разделительными символами. В R будем выделять начальными и конечными места.

Пример: $X = \{x, y, z\}$; $R = (z \vee x < y \vee z >)$; $R = | (| z | \vee | x | < | y | \vee | z | >) |$; 0 1 2 3 4 5 6 7 8 9 10

$R = x_{i1}, x_{i2}, \dots, x_{in}$, - произвольные регулярные выражения Тогда "а" связано \propto "b" можно перейти \in пом. любого числа: - непосредственно переходов; - переходов через символы, взятых по 1 разу в том порядке, в каком они входят в слово α - а - следует за местом $\alpha \Leftrightarrow$ всякое место а связано с местом b в слове а - Если от места а к месту b можно перейти с помощью одних лишь переходов, то есть если а связано с b - E(путем символов)

Лекция 9

3.2.3 Правила подчинения места в регулярном выражении
 $i j i j | \searrow |$ или $| \swarrow |$

- 1) Начальные места всех термов или букв или символов многочлена, помещенных в "()" или "< >" подчиняется

месту, располагающемуся слева открывающей скобки $(a \vee b) | (a | \vee | b |) | \searrow$

- 2) Место, располагающее справа от закрывающей скобки, подчинено конечным местам всех термов многочлена, заключенных в эти скобки. А в случае " $< >$ " еще и месту слева от открывающей скобки $| (| a | \vee | b |) | < | a | \vee | b | > | \cdot \searrow \searrow \searrow \searrow \searrow$
- 3) Начальные места всех термов многочлена заключены в " $< >$ ", подчиненных месту расположенных справа от закрывающей скобки $| < | a | \vee | b | > | \cdot \swarrow \swarrow$
- 4) Если место "с" подчиняется месту "b", а место "b" подчиняется месту "a", то место "с" подчиняется месту "a"
- 5) Каждое место подчиняется самому себе
- 6) Других случаев подчинения в регулярных выражениях нет.

X - алфавит (без вспомогательных символов) Основным местом называют место, слева от которого находится символ алфавита X, а также начальное место 0 1 2 3 4 5 $| (| a | \vee | b |) |$

Место справа будет предосновным (выше 1, 3)

3.2.4 Описание алгоритма синтеза конечного автомата по регулярному выражению

Пример следующее правило метки, синтеза автомата. Каждое состояние q_i определяется подмножество множества основных мест в регулярных выражения, отмечается множество, содержащее все те и только те выражения R_1 , конечные места которых подчиняются хотя бы одному основному месту из числа мест, входящему в подмножество q_i .

Некоторое слово $\alpha \in (X)^*$ переводит в S тогда (из q_0 в q_j), когда начальное место регулярного выражения R_i связано с конечным местом словом α

1. В заданных R_1, \dots, R_p все места различны
2. Каждому основному месту, в качестве его индекса предписывается N (некоторое число). При этом все начальным места предписывается индекс 0. Все введенные индексы, отмечаются основные места, называющиеся основными индексами.
3. Каждому основному индексу (a) распространяется в качестве неосновного индекса на все места, подчиняются

месту (а), но отличны от него самого. При этом каждому месту (р) получ. множество индексов.

4. Каждому состоянию синтез автомата S отмечается подмножество множества всех основных индексов, которые будем обозначать $q_i \rightarrow i_1 \vee i_2 \vee i_3 \dots, i_1, i_k$ - основные индексы. Пустое множество $\rightarrow ()$. Начальное состояние $q_0 \rightarrow \emptyset$. Таблица переходов формируется следующим образом. На пересечении X_i строки ($x_i \in X$) и g_j столбца ($g_j \in Q$) нашего множества индексов, отмеченное это состояние автомата, в числе которого находятся индексы всех тех, мест, в которых x_i следует за предосновными местами, и в числе индексов которых имеется хотя бы один индекс об. g_j . В противном случае, на пересечении имеется ().

5. $i_1 \vee i_2 \vee i_3 \dots$ Каждое подмножество отмеченных состояний КНА и соответствующий столбец отмеченный подмножество множества $R_1 \dots R_p$, конечно места которых содержит индекс $R_1 \dots_k$ **Пример:** $R_1 = \langle x \rangle \langle y \rangle$; $R_2 = \langle x \vee y \rangle$

$$1) R = \langle \langle x \rangle \vee \langle y \rangle \rangle, \langle \langle x \vee y \rangle \rangle \vee \langle x \rangle;$$

$$2) \quad (0) \quad 1 \quad 2 \quad (0) \quad 3 \quad 4 \quad 5$$

$$3) \quad |0| \quad 0 \quad |0| \quad 0 \quad 0 \quad 0 \quad 0$$

$$\cdot |1| |1| |1| (1) 3 3 3 \cdot |2| 2 4 4 4$$

$$\cdot (R_1) (R_1 \vee R_2) (R_1) (R_2) ()$$

X / Q	\emptyset	1 v 3 v 5	2 v 4	3 v 5	4
x	1 v 3 v 5	1 v 3 v 5	3 v 5	3 v 5	3 v 5
y	2 v 4	2 v 4	2 v 4	4	4

$$\emptyset \rightarrow q_1 \quad 1 \vee 3 \vee 5 \rightarrow q_2 \quad 2 \vee 5 \rightarrow q_3 \quad 3 \vee 5 \rightarrow q_4 \quad 4 \rightarrow q_5$$

$$(R_1) \rightarrow \emptyset \quad (R_1 \vee R_2) \rightarrow 1 \quad (R_2) \rightarrow 2 \quad () \rightarrow 3$$

$$\frac{\Delta}{\Delta} =$$

Q/Q	X	Y
q_1	$\frac{q_2}{1}$	$\frac{q_3}{0}$
q_2	$\frac{q_2}{1}$	$\frac{q_3}{0}$
q_3	$\frac{q_4}{2}$	$\frac{q_5}{3}$
q_4	$\frac{q_4}{2}$	$\frac{q_5}{3}$

Q/Q	X	Y
q_5	$\frac{q_4}{2}$	$\frac{q_5}{3}$

[[Pasted image 20250106175242.png]] $\alpha_1 = xy; \Rightarrow U = \emptyset; \alpha \in R_1$
 $\alpha_2 = xyx; \Rightarrow U = 2; \alpha \in R_2$ $\alpha_3 = yxy; \Rightarrow U = 3; \alpha \notin R_1, R_2$

Лекция 10

3.3 Минимизация конечного автомата

3.3.1 Необходимые определения $X^* U^* \rightarrow [S] \rightarrow$

$S_1 = \langle x_1, Q_1, U_1, \delta_1, \lambda_1 \rangle S_2 = \langle x_2, Q_2, U_2, \delta_2, \lambda_2 \rangle$

$g = \langle g_1, g_2, g_3 \rangle$

$\{g_1 : x_1 \rightarrow x_2; \{g_2 : Q_1 \rightarrow Q_2; \{g_3 : U_1 \rightarrow U_2;$

Гомоморфизм

$\forall x \in X_1; q \in Q_1, U \in U_1; \{\delta_2(g_1(x), g_2(q)) = g_2(\delta_1(x, q)); \} * \{ \lambda(g_1(x), g_2(q)) = g_2(\lambda_1(x, q)); \}$

Пример: $A = \{a_1, \dots, a_n\} B = \{b_1, \dots, b_n\} h_1 : A \rightarrow B; h_2 : B \rightarrow A; g : B \rightarrow B; \forall a \in A [h_2(g(a)) = g(h_1(a))];$ [[Pasted image 20250107180645.png]]

О:

S_1, S_2 называют изоморфными, если отображение g является изоморфизмом (взаимнооднозначными)

О

$S_1 = \langle X, Q_1, U, \delta_1, \lambda_1 \rangle S_2 = \langle X, Q_2, U, \delta_2, \lambda_2 \rangle \alpha = X_{i1}, X_{i2}$
 Если S_1 из $q' \in Q_1$ преобразует входное α в некоторое выходное $\beta U S_2$ из $q'' \in Q_2$ тоже α и β , то S_1 и S_2 соответственно называют неотличимыми

О

$\forall q' \in Q_1 S_1 \exists q'' \in Q_2 S_2, S_1$ и S_2 эквивалентны.

Замечание: Эквивалентные КНА означают реализацию одинакового отображения входных слов выходными словами.

Задача минимизации - это задача нахождения КНА с наименьшими и эквивалентным количеством состояний (к S_0) 1) $\forall S \exists S_0$ всегда существует единственный, с точностью до изоморфизма.
 2) Если количество состояний Q КНА S разбивается на 1

классов эквивалентными ($l \leq [Q]$), т.е. $Q_1 = \{q_{11}, q_{12}, \dots, q_{1i}\}, \dots, Q_l = \{q_{l1}, q_{l2}, \dots, q_{li}\}$. то $S_0 - |Q_0| = l$

Замечание: Таким образом, для неподходящих S_0 необходимо найти разбиение множеств состояний Q на классы эквивалентности.

3.3.2. Алгоритм минимизации конечного автомата (алгоритм мили) Алгоритм работает по шагам, на каждом шаге множество состояний Q разбивается на более мелкие подмножества

Лекция 11

3.3.2 Алгоритм Мили $S = \langle X, Q, U, \delta, \lambda \rangle$ $Q = \{q_1, q_2, \dots, q_n\}$;

Алгоритм работы по шагам. На i -ом шаге Q разбивается на более мелкие подмножества. - 1) $q', q'' \in Q \rightarrow Q_1$; Относим в один класс, если выполняется следующее соотношение $\forall x \in X : \lambda(q', x) = \lambda(q'', x)$ - $i+1$) $q', q'' \in Q_i \rightarrow Q_{i+1}$ $\forall x \in X : \delta(q', x), \delta(q'', x) \in Q_i$, l В конце каждого $i + 1$ шага: 1) $(i+1)$ шаг не изменяется $\forall j [Q_{i,j} = Q_{i+1,j}]$, то алгоритм заканчивает работу 2) Переход к следующему шагу $S: X = \{a, b, c\}; Q = \{q_1, \dots, q_9\} U = \{0, 1\}$

$\frac{\Delta}{\Delta} =$

Q / X	a	b	c
1	2/0	4/1	4/1
2	1/1	1/0	5/0
3	1/1	6/0	5/0
4	8/0	1/1	1/1
5	6/1	4/1	3/0
6	8/0	9/1	6/1
7	6/1	1/1	3/0
8	4/1	4/0	7/0
9	7/0	9/1	7/1

- 1) (1, 4, 6, 9) (2, 3, 8) (5, 7) Это классы эквивалентных состояний
- 2) (1, 4, 6) (2, 3, 8) (5, 7) (9)
- 3) (1, 4) (2, 3, 8) (5, 7) (9) (6)
- 4) (1, 4) (2, 8) (5, 7) (9) (6) (3)
- 5) —//— $Q_0 = \{q_1^0, q_2^0, q_3^0, q_4^0, q_5^0, q_6^0\}$

Q / X	a	b	c
1	2/0	1/1	1/1
2	1/1	1/0	3/0
3	5/1	1/1	6/0
4	3/0	4/1	3/1
5	2/0	4/1	5/1
6	1/1	5/0	3/0

ДЗ: 1) Сделать диаграмму переходов для исходного и мин. КНА
 2) Рассмотреть КНА, который синтезирован по т. Глушкова с результатом

3.4 Анализ конечных автоматов

3.4.1 Синтез КНА построение конечного автомата по описанию множества слов во входном алфавите, которые должны допускаться в этом входному алфавите

Анализ конечного автомата - обратная задача синтезу - получение описания множества входных слов по известному конечному автомату

Теорема Клини С. 1) Проблема синтеза Для любого регулярного выражения R всегда существует КНА, допускающий все множество входных слов (и только его), представимое этим регулярным выражением 2) Проблема анализа Любой КНА допускает такое множество входных слов, которое может быть представлено некоторым регулярным выражением R

Замечание: Множество входных слов, допускаемых КНА, получило название регулярного множества (языка) L(S)

3.4.2 Алгоритм анализа конечного автомата

(j) • номер такта = 0, 1, 2, ... ($x^j \in X$) - входной символ
 на j-ом такте ($q^j \in Q$) - состояние на j-ом такте ($u^j \in U$) - выходной символ на j-ом такте α - выходное слово
 НАЧАЛО

0) $j = 0$; $q^j = q_0$; $x^j = \emptyset$; - автомат стоит $u^j = \emptyset$;

1) $j = j + 1$

2) $q^j = \delta(x^j, q^{j-1})$

3) $u^j = \lambda(x^j, q^{j-1})$

4) $j \neq |\alpha| \rightarrow$ на п.1

5) если $u^j \in U_F$ $\alpha \in L(S)$ ДЗ:

- 6) Как изменится алгоритм, если КНА - частичный (заполнены не все состояния)?
 7) Как изменится алгоритм, если КНА неинициален?

3.4.3 $\frac{\triangle}{\triangle} =$

Q / X	X	Y
1	2/1	3/0
2	2/1	3/0
3	4/2	3/0
4	4/2	5/3
5	4/2	5/3

$$R_1 = \langle x \langle y \rangle R_2 = \langle x \vee y \rangle x$$

$$\alpha = xxxu;$$

j	x^j	q^j	u^j
0	-	1	-
1	x	2	1
2	x	2	1
3	x	2	1
4	y	3	\emptyset

$$\Rightarrow \alpha \in L(S)(R_1)$$

$$\alpha = xux \mid \mid x \mid q \mid u \mid \mid - \mid - \mid - \mid - \mid \mid 0 \mid - \mid 1 \mid 0 \mid \mid 1 \mid x \mid 4 \mid 2 \mid \mid 2 \mid y \mid 5 \mid 3 \mid \mid 3 \mid$$

Лекция 12

4.

4.1 Структурные модели лингвистических процессов

4.1.1 Определение формальной грамматики и языка

Формальной грамматикой G называется кортеж $\langle N, T, R, S \rangle$, где N - конечное непустое множество не терминальных, вспомогательных символов; T - конечное непустое множество терминальных (основных), причем $(N \cap T = \emptyset)$; R - конечное множество упорядоченных пар α, β ; $R = \{(\alpha, \beta)\} = \{\alpha \rightarrow \beta\}$; S - начальный символ аксиом для системы, $S \in N$

Замечание: $N = \{A, B, C, S\}$ $T = \{a, b, c\}$

О.

Продукции (правило вывода) грамматики G определяется следующим образом $R = \{\alpha \rightarrow \beta \mid \alpha \in (N \cup T)^* * N * (N \cup T)^*, \beta \in (N \cup T)^*\}$

-> отношение порядка - рефлексивность; - антисимметричность;
- транзитивность

Определение выводимости

Будем говорить, что из некоторого слова γ выводимо некоторое слово ϵ ($\gamma \Rightarrow_{\alpha} \epsilon$), причем каждое $\gamma, \epsilon \in (N \cup T)^*$

$\{\gamma = \delta_1 \alpha \delta_2 \mid \epsilon = \delta_1 \beta \delta_2 (\alpha \rightarrow \beta) \in R\}$

О.

Будем говорить, что из γ выводимо из ϵ , если существует такие $\gamma_0 = \gamma$ и $\gamma_n = \epsilon$ Последовательность называется выводом длины n $\gamma \Rightarrow_G^* \epsilon$ $\gamma_0 = \gamma$; $\gamma_n = \epsilon$ $\gamma_0 \Rightarrow_G \gamma_1 \Rightarrow_G \gamma_2 \dots \Rightarrow_G \gamma_n = \epsilon$;
 $\gamma_0, \gamma_1, \dots, \gamma_n$ - вывод n -малого

О.

Языком $L(G)$ порожденным грамматикой G называется следующее множество: $L(G) = \{\alpha \mid S \Rightarrow_G^* \alpha; \alpha \in T^*\}$

$\alpha \in T^*$, состоящих только из терминальных символов

О.

$\alpha \in L(G)$ - длина цепочки α ($|\alpha|$) **Пример:** $G = \langle T, N, R, S \rangle$ $T = \{a, b, c, *\}$ $N = \{S, A\}$ $R = \{S \rightarrow S * S, (1) S \rightarrow A + A, (2) S \rightarrow A, (3) A \rightarrow A * A, (4) A \rightarrow a\}$ (5) $S \Rightarrow^2 A + A \Rightarrow^4 A \cdot A + A \Rightarrow^5 a \cdot A + A \Rightarrow^5 a \cdot a + A \Rightarrow^5 a \cdot a + a$;

О.

D - помеченное дерево (граф вывода), если 1) корень помечен S , 2) D_1, \dots, D_n - поддереву, то корень каждого D_i помечен либо $A_i \in N$ (больше одной вершины у D_i), либо $a_i \in T$ (одна вершина у D_i) **Замечание:** Каждое D_i - дерево вывода в грамматике $G = \langle T, N, R, A_i \rangle$, в котором A_i - аксиома

Пример: $S A + A A * A a a a$

Свойство однозначности характеризует грамматику, а не язык, поскольку один и тот же язык может быть описан различными грамматиками

4.1.2 Классификация ФГ и языков

Распознающие Порождающие Преобразующие

О. Распознающие

позволяет ответить на вопрос, является ли цепочка правильной (принадлежит или нет?)

О. Порождающие

Позволяет строить любую правильную цепочку, давая при этомписание структуры, и не дает строить неправильные цепочки

О. Преобразующие

Для любой правильной цепочки, позволяет строить отображение её, задавая порядок реализации

Тип	Название	Вид нор. правил
3	Регулярные	$\alpha \in N; \beta \in T \cdot N$
2	КС (конкретно-свободные)	$\alpha \in N; \beta \in (T \cup N)^*$
1	МС, КЗ (конкретно-зависимые)	$\alpha \in (T \cup N)^* \cdot N \cdot (T \cup N)^* \beta \in (T \cup N)^*; \alpha \leq \beta $
\emptyset	Без ограничений	

Распознающая - известна цепочка ($a * a + a$), строим цепочку. Порождающая - используем грамматику, строим все цепочки. Прimitивно-рекурсивная функция и машина Тьюринга - это тип 0 **Пример:** $S \rightarrow aB$ $B \rightarrow bc$ (3) $c \rightarrow b$ ————— $S \rightarrow ABa$ $A \rightarrow cAAc$ (2) $B \rightarrow a$ ————— $AS \rightarrow ABC$ $Ab \rightarrow bbCb$ $B \rightarrow b$ $C \rightarrow c$

Диаграмма Эйлера - связь между типами. Также, как и в рекурсивных функциях. Если ФГ в центре, то одна и всех других типов.

Лекция 13

4.1.3 Формальные свойства грамматики

№	Название	Типы ФГ			
		3	2	1	0
1	$L_G = L(G) = \emptyset$	+	+	-	-
2	$ L(G) = \infty$	+	+	0	0
3	$L(G) = T^*$	+	-	-	-
4	$L(G_1) \leq L(G_2)?$	+	-	-	-
5	$L(G_1) \equiv L(G_2)?$	+	-	-	-
6	$L(G_1) \cap L(G_2) = \emptyset$	+	-	-	-
7	$\forall \alpha, \beta \in (T \cup N)^* : \alpha \Rightarrow \beta$	+	+	+	-
8		+	-	-	-
9		да	-	?	да

4.2 Автоматные грамматики в языке

T.

$G = \langle T, N, R, S \rangle \exists S = \langle A, Q, \delta, q_0, F \rangle$ 1) $T \rightarrow A$; 2) $N \rightarrow Q$; 3) $S \rightarrow q_0$ ($S \in N$; $q_0 \in Q$) 4) $[B \rightarrow aD] \rightarrow [\delta(a, B) = D]; [B \rightarrow a] \rightarrow [\delta(a, B) \in F]$ **и наоборот** **Пример:** $G: R = \{ S \rightarrow aB, B \rightarrow bS, B \rightarrow b \} \rightarrow^a (S) (B) \leftarrow^b$

$\frac{\triangle}{\triangle} =$

$S(q_0) \quad B(q_1)$	
a	B
b	S

4.2.1 $L^o \rightarrow S \rightarrow S^o \rightarrow G$ КНА минимизация конечные команды
ДЗ: $L = \{ a, b, c^k, d \mid m, n, k \in \mathbb{N} \}$ Реализовать цепочку

4.2.2 Подходы к реализации алгоритма

- 1) 1-й подход $G \rightarrow S \rightarrow \{ \alpha \in L(G) \mid \alpha \notin L(G) \}$
- 2) 2-й подход $G \rightarrow \{ \alpha \in L(G) \mid \alpha \notin L(G) \}$

4.3.1 Нормальные формы КС (конкретно-свободные) грамматики НФ (КС грамматика) НФ (Хомского) (НФ Грейбах)
. $G G$

O.

$G: \{A \rightarrow BC; . \{A \rightarrow a; . \{... \text{Справа либо 2 нетерминальных символа } (\in N), \text{ либо 1 терминальный символ } (\in T)$

О.

$G: \{A \rightarrow a\alpha; . \{A \in N_i; . \{a \in T; . \{ \alpha \in (N \cup T)^*\}$

Т.

$G: \exists G; G \ L(G) = L(G) = L(G) \ G \ \swarrow \searrow \ G \leftrightarrow G$ Т.е. существует 4 алгоритма реализации

4.3.2 Анализ КС грамматики

- 1) 1-й подход $G \rightarrow \{\alpha \in L(G) \uparrow \{\alpha \notin L(G) \ \alpha$
- 2) 2-й подход: $G \rightarrow G, \rightarrow S \rightarrow \{\alpha \in L(G) . \{\alpha \notin L(G)$

ДЗ: привести пример S (КНА с магазинной памятью) (рекомендовано в КС) - Реуэрд - Смит. Вводный курс

Лекция 14

[[Pasted image 20250107205755.png]]

5. Структурно-стохастические модели и их применение

$G = \langle N, T, R, S \rangle$

О.

$G_S = \langle N, T, R, S, S, S, R, R \rangle \ S = \{S_1, \dots, S_n\}$ - набор (множество) аксиом $|S| \geq 1$ S - мощность $S = \{p\}$ - вероятность ($p \in [0, 1]$) $S: S \rightarrow S$ - отображение множества аксиом $R = i_R: R \rightarrow R;$

О. $G_S^o - \langle N, T, R, S \rangle;$ - характеристическая грамматика По сути, она такая же, как и в обычной грамматике. **Пример:** $G_S = \langle N, T, R, S, R \rangle; |S| = 1; T = \{a\} \ N = \{S\} \ R = \{S \xrightarrow{0,8} aSa; . \{S \xrightarrow{0,2} aa$

$L(G_S) = \{\alpha \mid S \Rightarrow_{G_S} \alpha\} \Rightarrow L(G_S) = \{ \langle \alpha, P(\alpha) \rangle . \{\alpha \in T^* \ P(\alpha) = P(S \Rightarrow_{G_S}^* \alpha);$

О. СГ

$G_S \ L(G_S) \sum(P(\alpha)) = 1 \ \alpha \in h(G_S)$

$\alpha = aaaaaa \ S \Rightarrow^{0,8} aSa \Rightarrow^{0,8} aaSaa \Rightarrow^{0,2} aaaaaa \ P(\alpha) = 0,8 * 0,8 * 0,2 = \mathbf{0,128}$

5.2 Стохастический КНА

О. СКНА

Без выхода: $S_S = \langle X, Q, \delta_S, Q_F \rangle$ X - входной алфавит Q - алфавит внутренних состояний δ_S - начальное распределение состояний КНА $S \rightarrow S : \delta_S = \langle 0, 8^{q_0}; 0, 1^{q_1}; 0, 1^{q_2} \rangle$ δ_S - стохастическая функция переходов $\delta_S : X \cdot Q \rightarrow X \cdot Q \cdot Q \rightarrow \delta : X \cdot Q \cdot Q \rightarrow \delta$ Q_F - множество финальных состояний $Q_F \subseteq Q$ **Пример:** $(1) \xrightarrow{a} (2) \xrightarrow{b} (3) \xrightarrow{b} (4)$. $\cup^a \cup^b L(S) = \{a^n, b^m | m, n \leq 1\}$; $M =$

	1	2	3	4
1		a		
2		a	b	
3			b	b
4				

CA: $M = [[\text{Pasted image 20250107212510.png}]]$ $M(a) =$

	1	2	3	4
1		1		
2		0,4		
3				
4				

$M(b) =$

	1	2	3	4
1				
2			0,6	
3			0,6	
4				0,4

Замечание: Сумму элементов $M(ЗД)$ по слоям должна быть равна 1.

Т.

Любой однородный Марковский процесс (каждый переход должен иметь свою P) с дискретным временем и конечным количеством Q , может быть представлен, как язык, допускаемый СКНА

T.

Для любой СКНА: $CA S_S \exists$ СГ типа 3 G_S $L(S_S) = L(G_S)$ и наоборот
 СА - стохастический автомат СГ - стохастическая грамматика

Алгоритм построения СГ типа 3 по СА $S_s = \langle X, Q, S_S, Q_F \rangle$
 $G_S = \langle T, N, S, R, R \rangle$ 1) $T = X$; 2) $N = Q \setminus \{\#\} = Q \setminus Q_F$ 3)
 $[A_i \rightarrow^{P_i} x_j A_k] \Leftrightarrow [S_S(x_j q_i q_k) = P_{ik}]$ $[A_i \rightarrow x_j \dots] \Leftrightarrow [\delta_S(x_j q_i \#) = P_i]$ 4) $S \cong A_i \cong q_i$ $0 = \langle 01, 02, \dots, 0i, \dots \rangle$ **Пример:** $G_S = \langle T, N, R, S \rangle$; $T = \{a, b\}$; $S = \{S, A, B\}$

$R_S = \{S \xrightarrow{1} aA; . A \xrightarrow{0,4} aA; . A \xrightarrow{0,5} bB; . A \xrightarrow{0,1} b; . B \xrightarrow{0,6} bB; . B \xrightarrow{0,4} b\}$ $\{q_1 \rightarrow S \{q_2 \rightarrow A \{q_3 \rightarrow B$

ДЗ: Построить СКНА по данной грамматике

Пример: $\alpha = aabb$; $0 = \langle 1, 0, 0, 0 \rangle$. $F = \langle 0, 0, 0, 1 \rangle$. конечно
 $M(A) =$

	1	2	3	#
1		1		
2		0,4		
3				
#				

$M(B) =$

	1	2	3	#
1				
2			0,1	0,1
3			0,1	0,4
#				