МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

Д.А. ХОДАНОВИЧ, В.Н. СЕМЕНЧУК

ТЕОРИЯ АЛГОРИТМОВ И КОНЕЧНЫХ АВТОМАТОВ

Курс лекций по спецкурсу
«Теория алгоритмов и конечных автоматов»
для студентов математического и заочного факультетов
специальности 1-31 03 01 «Математика»
специализации «Алгебра и теория чисел»

Гомель 2011

УДК 519.14(075.8)
ББК 22.174 я73
Х 69

Рецензенты:
Скиба А.Н., профессор, доктор физико-математических наук, профессор кафедры алгебры и геометрии учреждения образования «Гомельский государственный университет имени Франциска Скорины»;
Новиков С.П., доцент, кандидат физико-математических наук, доцент кафедры высшей математики учреждения образования «Белорусский государственный университет транспорта»;
кафедра высшей математики учреждения образования «Гомельский государственный университет имени Франциска Скорины»

Рекомендовано к изданию научно-методическим советом учреждения образования «Гомельский государственный университет имени Франциска Скорины»

Ходанович Д.А., Семенчук, В. Н.
Х 69 Теория алгоритмов и конечных автоматов [Текст]: по спецкурсу «Теория алгоритмов и конечных автоматов» для студентов математического и заочного факультетов специальности 1-31 03 01 «Математика» специализации «Алгебра и теория чисел» / Д.А. Ходанович, В.Н. Семенчук; М-во образов. РБ, Гомельский государственный университет им. Ф. Скорины. – Гомель: ГГУ им. Ф. Скорины, 2011. – 37 с.

ISBN

Курс лекций по спецкурсу ставит своей целью оказание помощи студентам в усвоении основных теоретических положений теории алгоритмов и конечных автоматов и адресован студентам специальности 1-31 03 01 «Математика» специализации «Алгебра и теория чисел».

УДК 519.14(075.8)
ББК 22.174 я73

ISBN

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Дискретная математика – часть математики, которая зародилась в глубокой древности. Главной её спецификой является дискретность, т.е. антипод непрерывности. В широком смысле дискретная математика включает в себя и такие сложившиеся разделы математики, как теория чисел, алгебра, математическая логика, а также ряд разделов, которые наиболее интенсивно стали развиваться в середине 20-го века благодаря научно-техническому прогрессу, поставившему изучение сложных управляющих систем. В узком смысле дискретная математика ограничивается этими новыми разделами – теорией алгоритмов и теорией конечных автоматов. Именно эти разделы дискретной математики и рассматриваются в данном курсе.

Главной задачей курса лекций является овладение студентами методами и мышлением, характерными для современных разделов дискретной математики. Материал, вошедший в курс, знакомит студентов с теорией конечных автоматов и теорией алгоритмов. Данный курс лекций составлен таким образом, чтобы сократить число необходимых понятий до минимума и, с другой стороны, дать небольшое количество серьезных теорем.

В разделе конечные автоматы изучаются детерминированные и ограниченно-детерминированные функции.

В разделе элементы теории алгоритмов изучается строгое математическое определение теории алгоритмов, предложенное Тьюрингом – машина Тьюринга; примитивно-рекурсивные и частично-рекурсивные функции, а также устанавливается их связь с функциями вычислимыми по Тьюрингу.

ТЕМА 1 КОНЕЧНЫЕ АВТОМАТЫ

Автоматом называют дискретный преобразователь информации, способный принимать различные состояния, переходить под воздействием входных сигналов из одного состояния в другое и выдавать выходные сигналы.

Если множество состояний автомата, а также множество входных и выходных сигналов конечны, то автомат называют конечным автоматом. Все реальные автоматы являются конечными.

Информацию, поступающую на вход автомата, и преобразующую входную информацию принято кодировать конечной совокупностью символов. Эту совокупность называют алфавитом, отдельные символы, образующие алфавит, – буквами, а любые конечные упорядоченные последовательности букв данного алфавита – словами в этом алфавите.

Автоматы функционируют в дискретные моменты времени, которые обозначаются натуральными числами $t = 0, 1, 2, \ldots$ В каждый момент дискретного времени на вход автомата поступает один сигнал (буква), фиксируется определённое состояние автомата и с выхода снимается один сигнал. Реальные автоматы могут иметь, вообще говоря, несколько входов и выходов. В некоторых случаях для решения задач синтеза удобно заменить такие автоматы автоматами с одним входом и одним выходом. Для этого достаточно закодировать соответствующим образом входные и выходные сигналы исходного алфавита. Если, например, автомат имеет два входа, на каждый из которых подаются сигналы 0 или 1, то все возможные комбинации входных сигналов можно закодировать четырьмя буквами
(0, 0), (0, 1), (1, 0), (1, 1).

Процесс дискретного преобразования информации автоматами можно описать с помощью детерминированных функций.

1.1 Детерминированные функции

Обозначим через E^K={0, 1, …, k-1}, где k – некоторое натуральное число, а через E_K множество всех k-значных последовательностей *a* таких, что *a* = {*a*(1), *a*(2),…,*a*(t), …}, где *a*(i)$\in E^K$ для всех I = 1, 2,… .

Обозначим через P^K множество всех функций $y = f\ (x_1, x_2, \ldots, x_n)$, определённых на наборах $(a_1, a_2, \ldots, a_n)$, где $a_i \in E_K, i = 1,2,\ldots,n$ и принимающих значение из E_K. Функции из P^K преобразуют наборы k-значных последовательностей в k-значные последовательности. В множество P^K включим также все последовательности из E^K, рассматривая их как функции, зависящие от пустого множества переменных, т. е. как константы.

С помощью векторной записи функции от *n* переменных из P^K можно свести к функции от одной переменной. Обозначим набор переменных $(x_1, x_2, \ldots, x_n)$ через X, вместо $y = f(x_1, x_2, \ldots, x_n)$ будем писать $y = f$(X). При этом значение переменной X, есть вектор *a* = $(a_1, a_2, \ldots, a_n)$ компонентами которого являются последовательности из E_K,

$a_i = \{a_i(1), a_i(2), \ldots, a_i(t), \ldots\}, i = 1,2,\ldots,n$. Будем рассматривать *a* как последовательность векторов $a_i = \{a_i(1), a_i(2), \ldots, a_i(t), \ldots\}$, где

$a(i) = (a_1(i), a_2(i), \ldots, a_n(i)), i = 1,2,\ldots$.

Таким образом, мы будем считать, что выполняется тождество:

$$(\{a_1(1), a_1(2), \ldots, a_1(t), \ldots\}, \{a_2(1), a_2(2), \ldots, a_2(t), \ldots\}, \ldots, \{a_n(1), a_n(2), \ldots, a_n(t), \ldots\}) =$$

$$= \{(a_1(1), a_2(1), \ldots, a_n(1)), (a_1(2), a_2(2), \ldots, a_n(2)), \ldots, (a_1(t), a_2(t), \ldots, a_n(t)), \ldots\}.$$

Лемма 1.1 Число наборов $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, где $\alpha_i \in \{0,1,\ldots,k-1\}, i = 1,2,\ldots,n$ равно k^n.

Итак, функцию $y = f(x_1, x_2, \ldots, x_n) \in P^K$ с помощью векторной записи можно свести к функции $y =\ f(X) \in P^N$, где N $= k^n$. Таким образом, изучение функции $y = f(x_1, x_2, \ldots, x_n)$ из P^K можно свести к изучению функции от одной переменной из P^N, где N $= k^n$.

Определение 1.1 Функция $y = f(X)$ из P^N называется детерминированной, если каково бы ни было число *t* и каковы бы ни были последовательности *a* и *b* такие, что *a(1)=b(1), a(2)=b(2), … a(t)=b(t)* значения функций *α, β, где α=*$f(a)$, β=$f(b)$ представляют собой последовательности, у которых тоже совпадают первые *t* членов, т. е. α(1) = β(1), α(2) = β(2), …, α(t) = β(t).

Множество всех детерминированных функций обозначим через P_g^K.

Из определения детерминированной функции следует, что значение α*(t)* (α=$f(a)$) зависит только от значения первых *t* членов входной последовательности a, т. е. *a(1), a(2), …, a(t),* следовательно *α(t)=φ(a(1), a(2), …, a(t))*.

Приведём примеры как детерминированных, так и недетерминированных функций.

Пример 1.1 Рассмотрим функцию $y = f(x) \in P^2$, определённую следующим образом $f(a) = \begin{cases} (0,0\ldots), \text{если} \quad a = (0,0,\ldots) \\ (1,1,\ldots), \text{если} \quad a \neq (0,0,\ldots) \end{cases}$

Покажем, что данная функция недетерминированная. Действительно, возьмём две входные последовательности $a_1 = (0,0,0,\ldots)$ и $a_2 = (0,0,1,\ldots)$. Тогда $f(a_1) = (0,0,0,\ldots)$ и $f(a_2) = (1,1,1,\ldots)$. Следовательно, данная функция недетерминированная.

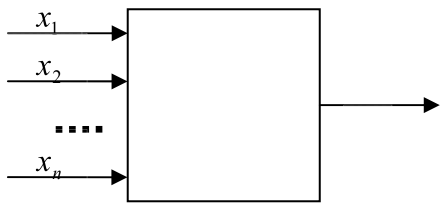
Пример 1.2 Рассмотрим функцию $f(x_1, x_2) = x_1 \cdot x_2$ из P^2, определённую следующим образом $f(x_1, x_2) = \{x_1(1) \cdot x_2(1), x_1(2) \cdot x_2(2), \ldots, x_1(t) \cdot x_2(t), \ldots\}$.Здесь выходная последовательность – почленная конъюнкция входных последовательностей. Очевидно, что $f(x_1, x_2) = x_1 \cdot x_2 \in P_g^{\,2}$.

Пример 1.3 Рассмотрим функцию z = x + y$\in P^2$, осуществляющую сложение 2-значных последовательностей в двоичной системе с бесконечным числом разрядов. Для этого используется обычный алгоритм сложения двух чисел столбиком

$$\begin{array}{r} \ldots x(3), x(2), x(1) \\ + \\ \ldots y(3), y(2), y(1) \\ \hline \ldots z(3), z(2), z(1). \end{array}$$

Очевидно, что *z(t)* определяется по первым *t* слагаемых, т. е. $x + y \in P_g^2$.

Детерминированная функция $y = f(x_1, x_2, \ldots, x_n)$ может быть проинтерпретирована следующим образом. Пусть мы имеем некоторый «дискретный преобразователь», в котором существует *n* входов $x_1, x_2, \ldots, x_n$ и один выход f.



На входы в моменты времени *t=1,2,…,m*, … подаются входные последовательности

$$a_1 = \{a_1(1), a_1(2), \ldots, a_1(t), \ldots\};$$
$$a_2 = \{a_2(1), a_2(2), \ldots, a_2(t), \ldots\};$$
$$\ldots$$
$$a_n = \{a_n(1), a_n(2), \ldots, a_n(t), \ldots\}.$$

И в эти же моменты *t* на выходе возникает выходная последовательность $\alpha = \{\alpha(1), \alpha(2), \ldots, \alpha(t), \ldots\}$, причем $\alpha = f(a_1, a_2, \ldots, a_n)$. Очевидно, что в дискретном преобразователе значения *α(t)* зависит только от значений входных последовательностей в момент времени *1,2,…,t* и не зависит от значений в будущие моменты времени. Поэтому преобразование f есть детерминированная функция.

1.2 Графическое задание детерминированных функций

Пусть $y = f(x_1, x_2, \ldots, x_n) \in P_D{}^K$. Выше мы показали, что с помощью векторной записи данную функцию можно свести к функции $y = f(X) \in P_D^N$, где $N = k^n$. Рассмотрим бесконечную фигуру (рисунок 1.1):

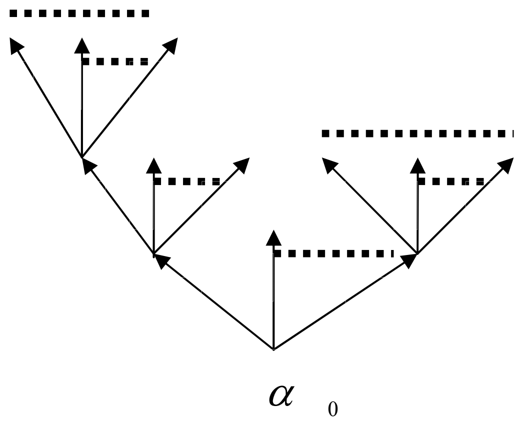


Рисунок 1.1

Называть её будем деревом, и построена она следующим образом. Возьмём произвольную вершину α_0, которую назовём корнем дерева. Из неё проведём N рёбер, которые образуют первый ярус. Из концов каждого из рёбер также проведём N рёбер, которые образуют второй ярус и т. д. Рёбра каждого пучка нумеруются слева направо числами 0, 1,…, N-1 или их значениями в k-ичной системе счисления.

В дальнейшем на рисунках номера рёбер будут опускаться. Далее, каждому ребру в построенном дереве произвольным образом припишем одно из чисел множества {0, 1,…, k-1}. В результате получим так называемое нагруженное дерево.

Рассмотрим следующее нагруженное дерево (рисунок 1.2):

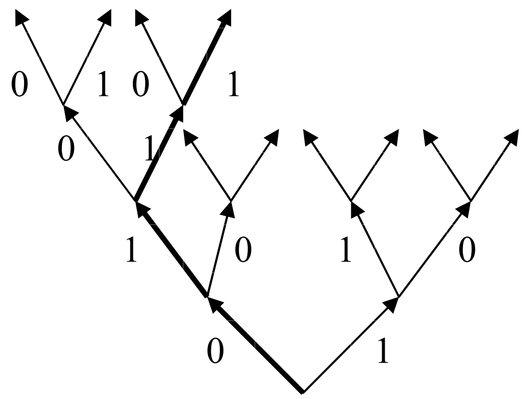


Рисунок 1.2

Начиная движение с корня дерева, пойдём по рёбрам. Так, например, последовательности (0, 0, 1, 1…), где числа 0, 0, 1, 1, … – номера рёбер, соответственно, 1-го,2-го,3-го,4-го и т. д. ярусов соответствует выделенный маршрут и последовательность (0, 1, 1, 1…).

Теорема 1.1 *Функция из* P^k *будет детерминированной тогда и только тогда, когда она может быть заданна с помощью нагруженного дерева.*

Доказательство. Покажем, что любое нагруженное дерево задает некоторую детерминированную функцию. Действительно, пусть $\{a(1), a(2), \ldots, a(t), \ldots\}$ – произвольная последовательность чисел, где $a(i) \in E^K$, I = 1, 2,…. Будем считать, что $a(1)$ – номер ребра 1-го яруса, $a(2)$ – номер ребра 2-го яруса и т. д. Данной последовательности в нагруженном дереве соответствует единственный маршрут, ведущий из корня дерева. Числа, приписанные выделенным ребрам образуют выходную последовательность $\{b(1), b(2), \ldots, b(t), \ldots\}$. Покажем, что построенная функция из P^K является детерминированной. Пусть $\{a_1(1), a_1(2), \ldots, a_1(t), \ldots\}$ и $\{a_2(1), a_2(2), \ldots, a_2(t), \ldots\}$ – две входные последовательности такие, что $a_1(1) = a_2(1), a_1(2) = a_2(2), \ldots, a_1(t) = a_2(t)$.

Ясно, что маршруты в нагруженном дереве, соответствующие данным последовательностям на первых t ярусах совпадают. А это значит, что $b_1(1) = b_2(1), b_1(2) = b_2(2), \ldots, b_1(t) = b_2(t)$, т. е. функция детерминированная. Обратное утверждение очевидно. Теорема доказана.

Рассмотрим следующие примеры:

Пример 1.4 $f(x) = \overline{x} \in P^2$. Ясно, что $f(x) = \overline{x} \in P_D^2$ и число ребер, выходящих из вершин равно $N = 2^1$. Построим дерево соответствующее данной функции (рисунок 1.3):

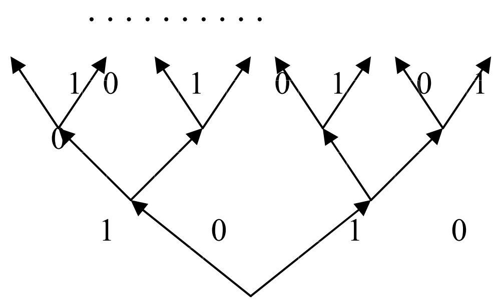


Рисунок 1.3

Например, входной последовательности {0, 1, 1, …} будет соответствовать входная последовательность {1, 0, 0, …}.

Пример 1.5 $f(x,y) = x \cdot y \in P_D^2$, которая задаётся следующим образом.

$f(x,y) = \{x(1) \cdot y(1), x(2) \cdot y(2), \ldots, x(t) \cdot y(t), \ldots\}$, где $x(t) \cdot y(t)$ – конъюнкция.

Для данной функции k=n=2 и число ребер, выходящих из вершин равно N = 2^2 = 4. Ребру с номером D = (0,0) соответствует значение (0,0) = 0

$$1 = (0,1) \quad 0 \cdot 1 = 0$$

$$2 = (1,0) \quad 1 \cdot 0 = 0$$

$$3 = (1,1) \quad 1 \cdot 1 = 1.$$

Следовательно, данной функции соответствует следующее нагруженное дерево (рисунок 1.4):

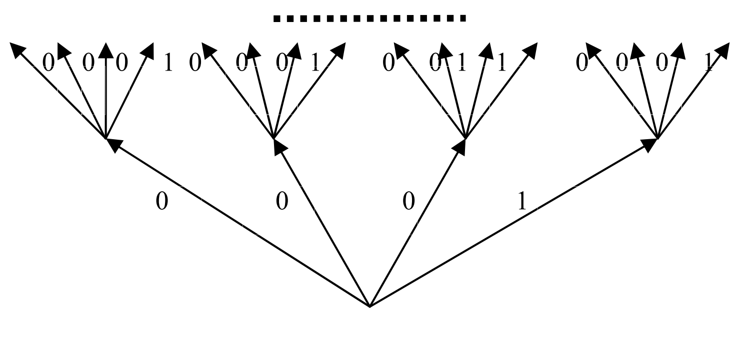


Рисунок 1.4

Пример 1.6 $f(z) = x + y \in P_D^2$, k = n = 1, N = 1^2 = 1.

$$z(t) = \begin{cases} x(t) = y(t) \pmod 2 \text{ \textit{при отсутствии переноса}} \\ x(t) + y(t) + 1 \pmod 2 \text{ \textit{при наличии переноса}} \end{cases}.$$

Дерево, соответствующее данной функции строится следующим образом. Процесс приписывания ребрам чисел начинается с 1-го яруса

0 = (0,0)   0+0=0

1 = (0,1)   0+1=1

2 = (1,0)   1+0=1

3 = (1,1)   1+1=0

При этом, если появляется перенос в следующий разряд, то конец соответствующего ребра кончается кружочком. Это позволяет выполнить вычисление в следующем ярусе (рисунок 1.5):

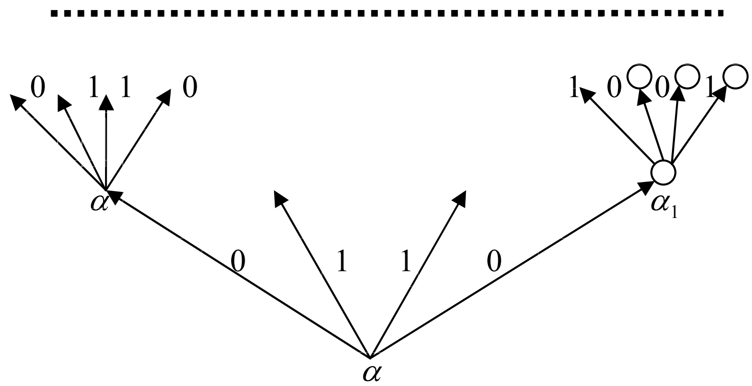


Рисунок 1.5

1.3 Ограниченно-детерминированные функции

Возьмем нагруженное дерево для некоторой детерминированной функции $f(X) = f(x_1, x_2, \ldots, x_n)$. Пусть α – произвольная его вершина n-го яруса. Данную вершину можно рассматривать как корень нагруженного дерева. Согласно теореме 1 оно определяет некоторую детерминированную функцию $f^{\alpha}(X)$.

Определение 1.2 Два поддерева с корнями α_1 и α_2 исходного дерева называются эквивалентными, если $f^{\alpha 1}(X) = f(X)^{\alpha 2}$.

Очевидно, что при естественном наложении двух эквивалентных поддеревьев их нумерации совпадают. Так, в дереве (рис. 1.3 и рис. 1.4) все поддеревья эквивалентны, а в дереве (рис. 1.5) поддеревья с корнями α эквивалентны, а с корнями α и α_1 не эквивалентны.

Определение 1.3 Весом дерева и весом соответствующей детерминированной функции называется максимальное число попарно неэквивалентных поддеревьев.

Например, все функции из примеров 1.4, 1.5 равны 1, а из примера 1.6 равны 2.

Определение 1.4 Детерминированная функция $f(x_1, x_2, \ldots, x_n)$ называется ограниченно – детерминированной функцией, если она имеет конечный вес.

Класс всех ограниченно – детерминированных функций обозначим через $P_{O.D.}^K$

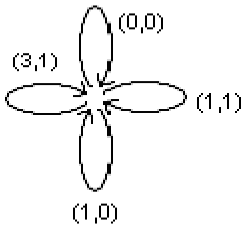
Функции из примеров 1.4, 1.5, 1.6 являются ограниченно-детерминированными функциями.

Рассмотрим следующую детерминированную функцию.

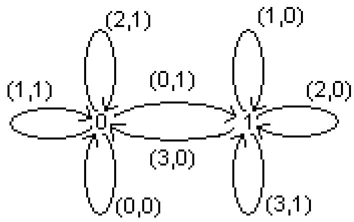
Пример 1.7 $f(x) = \{0,1,0,0,1,0,0,0,1,0,0,0,0,1\ldots\} \in P_D^2$. Ясно, что вес данной функции $r = \infty$, т. е. она не является ограниченно-детерминированной.

Пусть $f(X) \in P_{O.D.}^K$, вес которой равен r. Рассмотрим алфавит $Q = \{q(0), q(1), \ldots, q(r-1)\}$, который назовём внутренним алфавитом. Каждой вершине нагруженного дерева, соответствующей функции $f(X)$ припишем одну из букв алфавита Q с соблюдением следующего правила: эквивалентным вершинам приписываются одни и те же буквы из Q. В результате получаем так называемое полное нагруженное дерево.

Для любой ограниченно – детерминированной функции соответствующее ей полное нагруженное дерево можно свести к конечному дереву с занумерованными ребрами и вершинами. Если в нем провести отождествление эквивалентных вершин, то получим так называемую диаграмму Мура. В ней нулём отмечена начальная вершина и ребрам приписаны пары чисел (a, b), первое из которых обозначает номер ребра, а второе, чем данное ребро нагружено. Так функция $f(x, y) = x \cdot y \in P^2_{O.D.}$ соответствует диаграмме Мура.



А функция $z = x + y \quad \in P^2_{O.D.}$



1.4 Каноническое уравнение ограниченно-детерминированных функций

Пусть $y = f(x)$ – ограниченно-детерминированная функция с весом *r*.

Пусть $x = \{x(1), x(2), \ldots, x(t), \ldots\}$ – входная последовательность. Ей соответствует выходная последовательность $y = \{y(1), y(2), \ldots, y(t), \ldots\}$ и последовательность состояний $q = \{q(1), q(2), \ldots, q(t), \ldots\}$.

Возьмем другую входную последовательность $x' = \{x'(1), x'(2), \ldots, x'(t), \ldots\}$. Ей соответствуют, выходная последовательность и последовательность состояний

$y' = \{y'(1), y'(2), \ldots, y'(t), \ldots\}$;

$q' = \{q'(1), q'(2), \ldots, q'(t), \ldots\}$.

В общем случае из того, что $x(t) = x'(t)$ не следует, что $y(t) = y'(t)$. Однако, если $x(t) = x'(t)$ и $q(t-1) = q'(t-1)$, то $y(t) = y'(t)$ и $q(t) = q'(t)$. Другими словами это означает, что если два одноименных ребра ($x(t) = x'(t)$) выходят из эквивалентных вершин ($q(t-1) = q'(t-1)$), то они будут нагружены одной и той же буквой ($y(t) = y'(t)$) и будут входить в эквивалентные вершины ($q(t) = q'(t)$). Это означает, что

$$\begin{cases} y(t) = \Phi(x(t), q(t-1)) \\ q(t) = \Psi(x(t), q(t-1)) \end{cases} \tag{1.1}$$

Уравнения (1.1) называются каноническими уравнениями функции $y = f(x)$. Первое уравнение называется уравнением выход, второе уравнением перехода.

Уравнения (1.1) можно задать с помощью канонической таблицы.

x(t)	*q(t-1)*	*y(t)*	*q(t)*

Пусть x(t) и y(t) из $\{0,1\}$, а $q(t) \in Q, где\ |Q| = r$. Если вес r ≤ 2, то каноническая таблица есть таблица истинности. Если r > 2, то каноническая таблица не является таблицей истинности. Но с помощью кодирования всех чисел алфавита Q в двоичной системе счисления мы её можем преобразовать в таблицу истинности.

Рассмотрим теперь функцию от n переменных $y = f(x_1, x_2, \ldots, x_n) \in P^2_{O.D}$ с весом r > 1, $Q = \{0,1,2,\ldots,r-1\}$ – внутренний алфавит. Закодируем все числа из алфавита Q в двоичной системе счисления наборами из $\{0,1\}$ длинной $l = [\log_2 r] + 1$. В этом случае канонические уравнения искомой функции имеют вид:

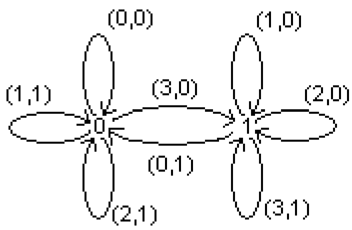
$$\begin{cases} y(t) = \Phi(x_1(t), x_2(t), \ldots, x_n(t), q_1(t-1), q_2(t-1), \ldots, q_l(t-1)) \\ q_1(t) = \Psi_1(x_1(t), x_2(t), \ldots, x_n(t), q_1(t-1), q_2(t-1), \ldots, q_l(t-1)) \\ \ldots \\ q_l(t) = \Psi_l(x_1(t), x_2(t), \ldots, x_n(t), q_1(t-1), q_2(t-1), \ldots, q_l(t-1)) \end{cases} \qquad (1.2)$$

В дальнейшем договоримся, что начальные состояния в канонических уравнениях (1.2) $q(0) = 0$, а в уравнениях (1.1) $q_1(0) = q_2(0) = \ldots = q_l(0) = 0$.

Пример 1.8 Найти канонические уравнения функции

$z = f(x, y) = x + y \in P^2_{O.D.}$

Ранее мы показали, что вес данных функций равен 2 и её диаграмма Мура



Построим каноническую таблицу.

x(t)	y(t)	q(t-1)	z(t)	q(t)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Данная каноническая таблица является таблицей истинности.

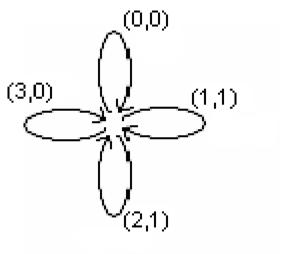
Запишем канонические уравнения.

$$\begin{cases} z(t) = \bar{x}(t)\bar{y}(t)q(t-1) \vee \bar{x}(t)y(t)\bar{q}(t-1) \vee x(t)\bar{y}(t)q(t-1) \vee x(t)y(t)q(t-1); \\ q(t) = \bar{x}(t)y(t)q(t-1) \vee x(t)\bar{y}(t)q(t-1) \vee x(t)y(t)\bar{q}(t-1) \vee x(t)y(t)q(t-1). \end{cases}$$

Используя законы алгебры логики:

$$\begin{cases} z(t) = x(t) + y(t) + q(t-1); \\ q(t) = y(t)q(t-1) \vee x(t)q(t-1) \vee x(t)q(t); \\ q(0) = 0. \end{cases}$$

Пример 1.9 Найти каноническое уравнение для функции заданной следующей диаграммой Мура



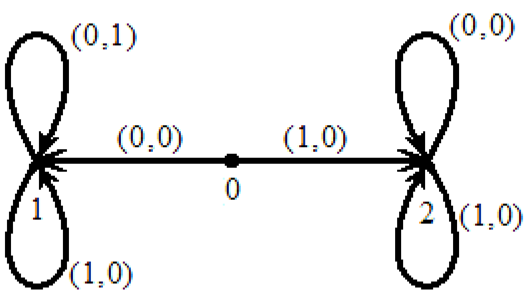
Строим каноническую таблицу.

x(t)	*y(t)*	*q(t-1)*	*z(t)*	*q(t)*
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	0

Отсюда $\begin{cases} z(t) = x(t) + y(t); \\ q(0) = 0. \end{cases}$

Заметим, что если вес функции равен 1, то в канонических уравнениях $q(t-1)$ будет отсутствовать.

Пример 1.10 Найти канонические уравнения ограниченно-детерминиро-ванной функции, заданной следующей диаграммой Мура**:**



Ясно, что вес данной функции равен 3. Построим каноническую таблицу для данной функции:

x(t)	*q(t-1)*	*y(t)*	*q(t)*
0	0	0	1
0	1	1	1
0	2	0	2
1	0	0	2
1	1	0	1
1	2	0	2

Данная таблица не является таблицей истинности. Преобразуем данную таблицу в таблицу истинности. Для этого значения второго и четвёртого столбца закодируем в двоичной системе счисления:

x(t)	$q_1(t-1)$	$q_2(t-1)$	*y(t)*	$q_1(t)$	$q_2(t)$
0	0	0	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	Не определена		
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	Не определена		

Доопределим данную функцию следующим образом:

x(t)	$q_1(t-1)$	$q_2(t-1)$	y(t)	$q_1(t)$	$q_2(t)$
0	0	0	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	1	1	1

Составим канонические уравнения, используя аппарат булевой алгебры:

1) *y(t)*= $\overline{x(t)}\;\overline{q_1(t-1)}\;q_2(t-1) \vee \overline{x(t)}\;q_1(t-1)q_2(t-1) \vee x(t)q_1(t-1)q_2(t-1) =$

$$= \overline{x(t)}\;\overline{q_1(t-1)}q_2(t-1) \vee q_1(t-1)q_2(t-1)(\overline{x(t)} \vee x(t)) =$$

$$= \overline{x(t)}\;\overline{q_1(t-1)}q_2(t-1) \vee \vee\, q_1(t-1)q_2(t-1) = q_2(t-1)(\overline{x(t)}\;\overline{q_1(t-1)} \vee q_1(t-1))\,;$$

$$2)\; q_1(t) = (x(t) \vee q_1(t-1) \vee q_2(t-1))(x(t) \vee q_1(t) \vee \overline{q_2(t-1)})\cdot$$
$$\cdot(\overline{x(t)} \vee q_1(t-1) \vee \overline{q_2(t-1)}) = (x(t) \vee q_1(t-1))(\overline{x(t)} \vee q_1(t-1) \vee \overline{q_2(t-1)}) =$$
$$= q_1(t-1) \vee x(t)(\overline{x(t)} \vee \overline{q_2(t-1)}) = q_1(t-1) \vee x(t)\overline{q_2(t-1)}\,;$$

$$3)\; q(t) = (x(t) \vee \overline{q_1(t-1)} \vee q_2(t-1))(\overline{x(t)} \vee q_1(t-1) \vee q_2(t-1))(\overline{x(t)} \vee$$
$$\vee\, \overline{q_1(t-1)} \vee q_2(t-1)) = (\overline{q_1(t-1)} \vee q_2(t-1))(\overline{x(t)} \vee q_1(t-1) \vee q_2(t-1)) =$$
$$= q_2(t-1) \vee \overline{q_1(t-1)}(\overline{x(t)} \vee q_1(t-1)) = q_2(t-1) \vee \overline{q_1(t-1)x(t)}.$$

Итак, искомые канонические уравнения имеют вид:

$$\begin{cases} y(t) = q_2(t-1)(\overline{x(t)} \vee q_1(t-1)); \\ q_1(t) = q_1(t-1) \vee x(t)\overline{q_2(t-1)}; \\ q_2(t) = q_2(t-1) \vee \overline{q_1(t-1)}x(t); \\ \qquad q_1(0) = q_2(0) = 0. \end{cases}$$

Каждой ограниченно-детерминированной функции можно сопоставить канонические уравнения. Однако выбор канонических уравнений не однозначен. Эта неоднозначность связана:

1) с различными способами кодирования состояний;

2) с различными способами доопределения функций.

Очевидно, что канонические уравнения позволяют вычислить
по входной последовательности $a=\{a(1),a(2),\ldots,a(\mathrm{t}),\ldots\}$
выходную последовательность $b=\{b(1),b(2),\ldots,b(\mathrm{t}),\ldots\}$.

Итак, для задания конечного автомата фиксируется три конечных множества (алфавита):

– множество возможных входных сигналов $x = (x_1, x_2, ..., x_m)$;

– множество возможных выходных сигналов $y = (y_1, y_2, ..., y_k)$;

– множество возможных внутренних состояний автомата $Q = (q_0, q_1, ..., q_r)$.

На этих множествах задаются две детерминированные функции:

– функция переходов Ψ, определяющая состояние автомата $q(t)$ дискретного времени t в зависимости от состояния автомата $q(t-1)$ и значения входного сигнала в момент времени t: $q(t)= \Psi(x(t),q(t-1))$;

– функция выходов Φ, определяющая зависимость выходного сигнала автомата $y(t)$ от состояния автомата $q(t-1)$ и входного сигнала $x(t)$ в момент времени t: $y(t)=\Phi(x(t),q(t-1))$.

Кроме того, на множестве состояний автомата фиксируется одно из внутренних состояний $q(0)$ в качестве начального состояния.

1.5 Вопросы для самоконтроля

1 Дайте определение детерминированной функции.

2 Приведите примеры детерминированных функций.

3 Приведите примеры недетерминированных функций.

4 Приведите графическую интерпретацию детерминированных функций.

5 Что такое бесконечное нагруженное дерево?

6 Что такое вес бесконечно нагруженного дерева?

7 Какие функции называются ограниченно-детерминированными?

8 Приведите примеры ограниченно-детерминированных функций.

9 Приведите примеры неограниченно-детерминированных функций.

10 Что такое диаграмма Мура?

11 Дайте определение канонических уравнений ограниченно-детерминированных функций.

ТЕМА 2 ЭЛЕМЕНТЫ ТЕОРИИ АЛГОРИТМОВ

Понятие алгоритма стихийно формировалось с древнейших времен. Современный человек понимает под алгоритмом четкую систему инструкций о выполнении в определенном порядке некоторых действий для решения всех задач какого-то данного класса.

Многочисленные и разнообразные алгоритмы окружают нас буквально во всех сферах жизни и деятельности.

Большое количество алгоритмов встречается при изучении математики буквально с первых классов школы. Это, прежде всего, алгоритмы выполнения четырех арифметических действий над различными числами – натуральными, целыми, дробными, комплексными. Примерами известных алгоритмов являются алгоритм Евклида нахождения наибольшего общего делителя двух натуральных чисел, вычисление определителей различных порядков, вычисление рангов матриц с рациональными элементами, приближенное вычисление корней уравнений и т.д.

В начале 20 века у математиков начали возникать подозрения в том, что некоторые массовые задачи, по-видимому, не имеют алгоритмического решения. Для точного доказательства несуществования алгоритма необходимо иметь его точное математическое определение. Первые работы по уточнению понятия алгоритма и его изучению были выполнены в 1936–1937 гг. А. Тьюрингом, Э. Постом, К. Геделем, А.А. Марковым, А. Черчем. Было выработано несколько определений понятия алгоритма, но впоследствии выяснилось, что все они равносильны между собой, то есть определяют одно и то же понятие.

2.1 Машина Тьюринга

Машина Тьюринга есть математическая (воображимая) машина, а не машина физическая. Она есть такой же математический объект, как функция, производная, интеграл и т.д. А также, как и другие математические понятия, понятие машина Тьюринга отражает объективную реальность, моделирует некие реальные процессы.

Машина Тьюринга состоит из ленты, управляющего устройства и считывающей головки.

Лента разбита на ячейки. Во всякой ячейке в каждый момент времени находится в точности один символ из внешнего алфавита $A=\{a_0,a_1,\ldots a_{n-1}\}$, $n\geq 2$. Некоторый символ алфавита A называется пустым, любая ячейка, содержащая в данный момент пустой символ, называется пустой ячейкой.
В дальнейшем в качестве внешнего алфавита будем использовать $A=\{0,1\}$, где в качестве пустого символа будем использовать 0 (нуль). В каждый момент времени лента содержит конечное число ячеек, но в процессе работы машины можно пристраивать новые ячейки в пустом состоянии.

Управляющее устройство в каждый момент времени находит в некотором состоянии q_i, принадлежащее множеству $Q\{q_0,q_1,\ldots,q_{r-1}\}$, $r\geq 1$. Множество Q называется внутренним алфавитом. В дальнейшем начальное состояние будем обозначать символом q_1, а заключительное символом q_0.

Считывающая головка перемещается вдоль ленты так, что в каждый момент времени она обозревает ровно одну ячейку ленты. Головка может считывать содержимое обозреваемой ячейки и записывать в нее вместо обозреваемого символа некоторый новый символ из внешнего алфавита.

Работа машины Тьюринга определяется программой. Программа состоит из команд. Каждая команда представляет собой выражение одного из следующего вида:

1) $q_i a_j \rightarrow q_k a_e$;
2) $q_i a_j \rightarrow q_k a_e R$;
3) $q_i a_j \rightarrow q_k a_e L$.

Команда 1 заключается в том, что содержимое a_j обозреваемой на ленте ячейки стирается, а на его место дописывается символ a_e (который может совпадать с a_j), машина переходит в новое состояние q_k (оно может совпадать с предыдущим состоянием q_i).

Команда 2 работает аналогично команде 1, и дополнительно сдвигает считывающую головку в соседнюю справа ячейку.

Команда 3 работает аналогично команде 1, и дополнительно сдвигает считывающую головку в соседнюю слева ячейку.

Если считывающая головка находится в крайней справа (слева) ячейки ленты и происходит ее сдвиг вправо (влево), то к ленте пристраивается новая ячейка в пустом состоянии.

Машинным словом или конфигурацией называется слово вида

$$a_{i_1}, a_{i_2}, \ldots, q_k a_{i_t}, \ldots, a_{i_k},$$

где $a_{i_j} \in$ A, $q_k \in$ Q. Символ q_k пишется перед символом обозреваемой ячейки. Причем символ q_k может быть самым левым, но не может быть самым правым. В дальнейшем будем использовать следующее обозначение: a_i^n обозначает слово $a_i\ a_i\ \ldots a_i = a_i^n$

Например, конфигурация $1^3 q_1 0^2 1^2$ на ленте выглядит следующим образом: ▼q_1

1	1	1	0	0	1	1

Машина Тьюринга считается заданной, если заданы программа, внешний и внутренний алфавиты, и указано, какие из символов обозначают начальное и заключительное состояние.

Если машина Тьюринга выходит на заключительное состояние, то она называется остановившейся.

Пусть машина Тьюринга начинает работать в некоторый (начальный) момент времени. Слово, записанное в этот момент на ленте, называется начальным. Чтобы машина Тьюринга начала работать, необходимо поместить считывающую головку против какой-либо ячейки на ленте и указать, в каком состоянии машина Тьюринга находится в данный момент.

Если P – начальное слово, то машина Тьюринга, начав работу ”на слове” P либо остановится через определенное число шагов, либо никогда не остановится. В первом случае говорят, что машина Тьюринга применима к слову Р и результатом применения машины к слову Р является слово М, соответствующее заключительной конфигурации.

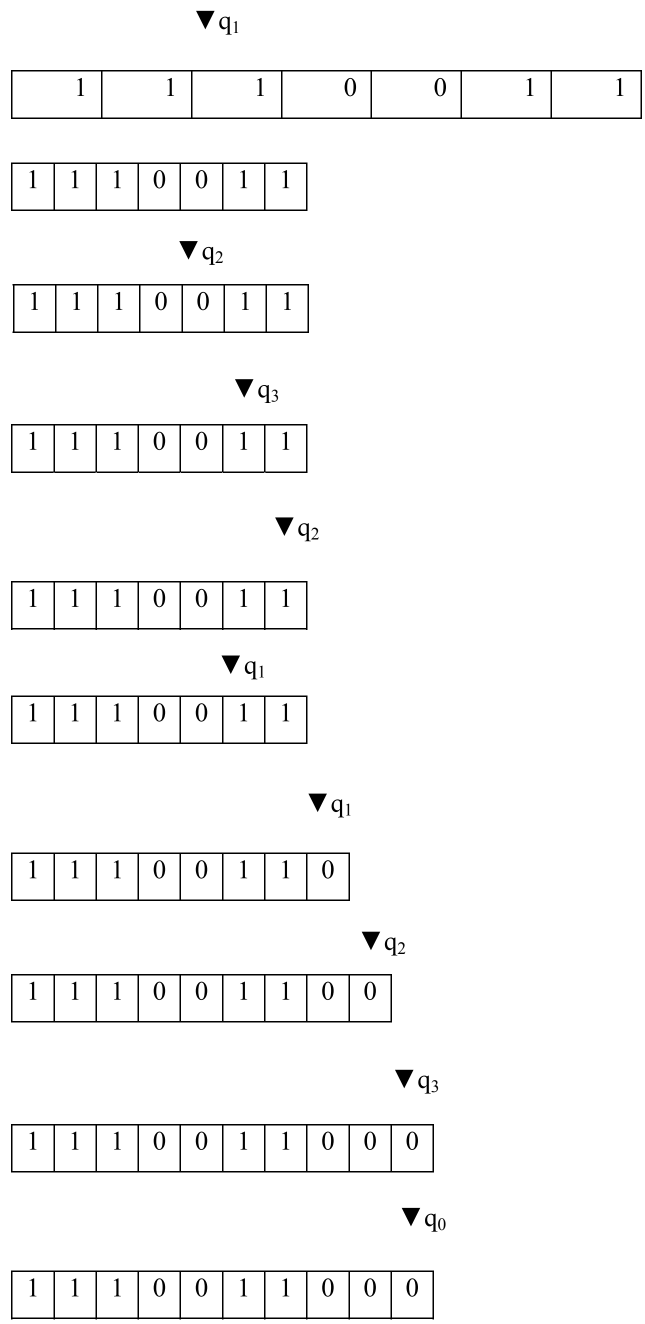
Во втором случае говорят, что машина не применима к слову Р.

Пример 2.1 Выяснить, применима ли машина Тьюринга, задаваемая следующей программой

$q_1 0 \to q_2 0\ R$, $q_1 1 \to q_1 1\ R$, $q_2 0 \to q_3 0\ R$, $q_2 1 \to q_1 1\ L$, $q_3 0 \to q_0 0$, $q_3 1 \to q_2 1\ R$, к слову $P = q_1\, 1^3 0^2 1^2$.

▼q_1

$q_1 1 \to q_1 1\ R$	$1^3\, q_1 0^2 1^2$
$q_1 0 \to q_2 0\ R$	$1^3 0\, q_2 0 1^2$
$q_2 0 \to q_3 0\ R$	$1^3 0^2\, q_3 1^2$
$q_3 1 \to q_2 1\ R$	$1^3 0^2 1\, q_2 1$
$q_2 1 \to q_1 1\ L$	$1^3 0^2\, q_1 1^2$
$q_1 1 \to q_1 1\ R$	$1^3 0^2 1^2\, q_1 0$
$q_1 0 \to q_2 0\ R$	$1^3 0^2 1^2\, 0 q_2 0$
$q_2 0 \to q_3 0\ R$	$1^3 0^2 1^2\, 0^2 q_3 0$
$q_3 0 \to q_0 0$	$1^3 0^2 1^2\, 0^2 q_0 0$



Следовательно, данная машина Тьюринга применима к слову Р и заключительная конфигурация имеет вид: $1^3 0^2 1^2 0^2 q_0 0$.

В дальнейшем запись $M \underset{T}{\Rightarrow} M_1$, будет означать, что машина Тьюринга (T) через конечное число циклов перерабатывает машинное слово M в машинное слово M_1.

Функция называется вычислимой по Тьюрингу, если существует машина Тьюринга, вычисляющая её.

Во-первых, напомним, что речь идёт о частично-числовых функциях. Во-вторых, договоримся, что значения $x_1, x_2, ..., x_n$ аргументов функции $f\,(x_1, x_2 \ldots x_n)$ на ленте будут записываться следующим образом:

$$0\underbrace{11\ldots11}_{x_1}\;0\underbrace{11\ldots1}_{x_2}\;0\ldots0\underbrace{11\ldots1}_{x_n}\;0,$$

или $01^{x_1}01^{x_2}0...01^{x_n}0$.

Начинать переработку данного слова будем из стандартного положения, то есть из положения, при котором в состоянии q_1 обозревается крайняя правая единица описанного слова. Если функция $f\,(x_1, x_2 \ldots x_n)$ определена на данном наборе значений аргументов, то в результате на ленте должно быть записано подряд $f\,(x_1, x_2 \ldots x_n)$ единиц, в противном случае, машина должна работать бесконечно. При выполнении всех перечисленных условий, будем говорить, что машина Тьюринга вычисляет данную функцию f.

Пример 2.2 Построить машину Тьюринга, вычисляющую функцию O(x) = 0. Для этого надо сконструировать машину Тьюринга (T), (т.е. составить программу) которая начинает работу со слова $q_1 01^x$ (x – любое натуральное число) останавливается, когда на ленте нет единиц

$q_1 01^x \Rightarrow q_0 0$

Данная программа имеет вид:

$q_1 0 \to q_2 0R$

$q_2 0 \to q_0 O$

$q_2 1 \to q_2 0R$

Пример 2.3 Построить машину Тьюринга, вычисляющую функцию: $f(x,y) = x + y$. Построим машину Тьюринга (Т), которая, начиная работу со слова $q_1 01^x 01^7$ останавливается когда на ленте $x+y$ единиц

$q_1 01^x 01^7 \Rightarrow q_0 01^{x+y}$

$q_1 0 \to q_2 0R$	$q21^x 01^y$
$q_2 0 \to q_0 OR$	$q_0 01^y$
$q_2 1 \to q_3 1$	$q_3 1^x 01^y$
$q_3 1 \to q_3 1R$	$1^x q_3 01^y$
$q_3 0 \to q_4 1$	$1^x q_4 1^{y+1}$
$q_4 1 \to q_4 1L$	$q_4 01^{x+y+1}$
$q_4 0 \to q_5 0R$	$q_5 1^{x+y+1}$
$q_5 1 \to q_0 0$	$q_0 01^{x+y}$

Пример 2.4 Построить машину Тьюринга, вычисляющую функцию $f(x) = 2/x$. Эта функция не всюду определена. Лишь при $x = 1$, $x = 2$, её значениями являются натуральные числа 2 и 1 соответственно. Поэтому можно построить машину Тьюринга, которая начинает работу либо со слова $q_1 01$, либо со слова $q_1 01^2$ останавливается, когда на ленте две единицы, одна единица соответственно. В остальных случаях машина начина-

ет работу со слов $q_2 01^x$ $(x \neq 1,\ x \neq 2)$ работает бесконечно. Такая машина задаётся следующей программой:

$q_2 0 \to q_2 0R$	$q_2 1^x$
$q_2 0 \to q_2 0$	$x = 0$
$q_2 1 \to q_3 1R$	$1\, q_3 1^{x-1}$
$q_3 0 \to q_0 1L$	$q_0 1^2,\ x = 2$
$q_3 1 \to q_4 1R$	$1^2\ q_4 1^{x-2}$
$q_4 1 \to q_4 1$	$x > 2$
$q_4 0 \to q_5 0L$	$1\, q_5 1$
$q_5 1 \to q_0 0L$	$q_0 1,\ x = 2$

2.2 Рекуррентные функции

В дальнейшем под множеством натуральных чисел N будем понимать множество $N = \{0,1,2,\ldots,k,\ldots\}$

Пусть $y = f(x_1, x_2,\ldots, x_n)$ – функция от n переменных. Обозначим $D(y)$ – область определения функции $y = f(x_1, x_2,\ldots, x_n)$, $E(y)$ – область значений функции $y = f(x_1, x_2,\ldots, x_n)$.

Функция $y = f(x_1, x_2,\ldots, x_n)$ называется числовой функцией, если:

1) $D(y)=N \times\cdot N \cdot\times \ldots \times\cdot N = N^n$;

2) $E(y) \subseteq N$

Функция $y = f(x_1, x_2,\ldots, x_n)$ называется частично числовой функцией, если:

1) $D(y) \subseteq N \times\cdot N\cdot\times\ldots\times\cdot N = N^n$;

2) $E(y) \subseteq N$.

Следующие числовые функции мы будем называть простейшими:

1) $O(x) = 0$ – нуль-функция

2) $I^n_m(x_1, x_2,..., x_n) = x_m$, $1 \leq m \leq n$ – функция повторяющая значение своих аргументов;

3) $S(x) = x+1$ – функция следования.

Определим следующие три операции: суперпозиции, примитивной рекурсии и минимизации.

Операция суперпозиции

Будем говорить, что n – местная функция φ получается из m – местной функции ψ и n – местных функций $f_1,f_2,...,f_m$ с помощью операции суперпозиции, если для всех $x_1,x_2,...,x_n$ справедливо равенство:

$$\varphi(x_1,x_2,...,x_n) = \psi(f_1(x_1, x_2,..., x_n),..., f_m(x_1, x_2,..., x_n))$$

Операция примитивной рекурсии

Будем говорить, что $(n+1)$ – местная функция f получается из n – местной функции g и $(n+2)$ – местной функции h с помощью операции примитивной рекурсии, если при любых значениях $x_1, x_2,..., x_n$ выполняются равенства:

$$f(x_1, x_2,..., x_n, 0) = g(x_1, x_2,..., x_n)$$

$$f(x_1, x_2,..., x_n, 1) = h(x_1, x_2,..., x_n, 0, f(x_1, x_2,..., x_n, 0))$$

- -

$$f(x_1, x_2,..., x_n, y+1) = h(x_1, x_2,..., x_n, y, f(x_1, x_2,..., x_n,y))$$

Эти равенства называют схемой примитивной рекурсии. И тот факт, что функция f получается из функций g, h с помощью операции примитивной рекурсии, записывается следующим образом: $f=R(g,h)$.

Определение 2.1 Функция f называется примитивно рекурсивной функцией, если она получается из простейших функций с помощью операций суперпозиции и примитивной рекурсии, взятых конечное число раз в любой последовательности.

Из данного определения следует, что любая примитивно рекурсивная функция является числовой функцией.

Множество всех примитивно рекурсивных функций обозначим через $P_{п.р.}$

Пример 2.5 Доказать, что функция $f(x,y)= x+y$ примитивно рекурсивна.

Действительно. Справедливы следующие тождества

$f(x,0) =x+0 = x = g(x)$

$f(x,y+1) = x+(y+1) = (x+y)+1 = f(x,y)+1$

Отсюда следует, что $x+y = R(g(x) = x,\ h(x,y,z) = z+1)$. Так как функции g, h – простейшие функции, то $x + y \in P_{п.р.}$

Пример 2.6 Доказать, что функция $f(x,y) = x{\cdot}y$ примитивно рекурсивна.

Действительно. Справедливы следующие тождества:

$f(x,0) = x{\cdot}0 = 0 = g(x)$

$f(x,y+1) = x(y+1) = xy+x = f(x, y) + x$

Отсюда следует, что

$x+y = R(g(x) = 0,\ h(x,y ,z) = z+ x)$

Как следует из примера 1 функция $h(x,y,z) = x+z \in P_{п.р.}$ А это значит, что $xy \in P_{п.р.}$

Рассмотрим функцию $x \dot{-} y = \begin{cases} x - y, если\ x \geq y; \\ 0, если\ x < y. \end{cases}$

Данную функцию называют усечённой разностью.

Пример 2.7 Показать, что функция $f(x,y) = x \dot{-} y$ примитивно рекурсивна.

В начале заметим, что функция $x \dot{-} 1$ примитивно рекурсивна. Действительно:

$0 \dot{-} 1 = 0 = g(x)$

$(x+1) \dot{-} 1 = x = h(x,y)$

Следовательно, $x \dot{-} 1 = R(g(x) = 0,\ h(x,y) = x)$. Итак, $x \dot{-} 1 \in P_{п.р.}$

Далее, нетрудно показать, исходя из определения усечённой разности, что эти функции удовлетворяют также равенствам:

$$x \dot{-} 0 = x = g(x)$$

$$x \dot{-} (y+1) = (x \dot{-} y) \dot{-} 1 = h(x, y, x \dot{-} y)$$

для любых x и y. Данные тождества показывают, что

$$x \dot{-} y = R(g(x) = 0,\ h(x,y,z) = z \dot{-} \alpha).$$

Так как функция $h(x,y,z) = z \dot{-} \alpha \in P_{n.p.}$, то $x \dot{-} y \in P_{n.p.}$

Так как любая примитивно рекурсивная функция является числовой функцией, то, очевидно, что $x - y \notin P_{n.p.}$

Пример 2.8 Покажем, что $|x-y|$– примитивно рекурсивная функция.

Действительно. Нетрудно показать, что $|x-y|=(x \dot{-} y)+(y \dot{-} x)$. Теперь полученный результат следует из примера 5 и примера 7.

Операция минимизации. Будем говорить, что n-местная функция $g(x_1,x_2,\dots,x_n)$ полученная из $(n+1)$-местной функции $f(x_1,x_2,\dots,x_n,y)$ с помощью операции минимизации или оператора наименьшего числа, если для любых $x_1, x_2,\dots, x_n, y$ равенство $g(x_1,x_2,\dots,x_n) = y$ выполняется тогда и только тогда, когда:

1) $f(x_1,x_2,\dots,x_n,t)$ определено и $f(x_1,x_2,\dots,x_n,t) > 0$ для любых $0 \le t < y$;

2) $f(x_1,x_2,\dots,x_n,y) = 0$.

Если какое-либо из условий 1), 2) будет невыполнено, то функция $g(x_1,x_2,\dots,x_n)$ не определена при наборе $x_1, x_2,\dots, x_n$.. Короче говоря, величина $g(x_1,x_2,\dots,x_n)$ равна наименьшему значению аргумента y, при котором выполняется последнее равенство.

Используется следующее обозначение:

$$g(x_1,x_2,\dots,x_n) = M_y\,[f(x_1,x_2,\dots,x_n,y)=0].$$

Про функцию g говорят, что она получена из функции f при помощи операции минимизации.

Определение 2.2 Функция f называется частично рекурсивной функцией, если она может быть получена из простейших функций с помощью операции суперпозиции, примитивной рекурсии и минимизации, взятых конечное число раз в любой последовательности.

Класс частично рекурсивных функций обозначим P_{rp}.

Обозначим через P_p – класс рекурсивных функций, т.е. всех числовых функций из P_{rp}.

Пример 2.9 Доказать, что частично числовая функция $g(x) = \frac{x}{2}$ частично рекурсивна.

Вначале заметим, что данная функция получается из функции $f(x, y) = |x - 2y|$ с помощью операции минимизации, т.е. $\frac{x}{2} = M_y(|x - 2y| = 0)$.

Согласно примерам 2 и 4 функция $f(x, y) = |x - 2y|$ примитивно рекурсивна. А это значит, что $g(x) = \frac{x}{2}$ – частично рекурсивная функция.

Данный пример показывает, что класс P_{rp} существенно шире, чем класс P_p. Можно сказать, что и класс P_p существенно шире, чем класс P_{np}, т.е. P_{np} С P_p С P_{rp}.

2.3 Тезисы Тьюринга и Чёрча

Одно из основных свойств алгоритма заключается в том, что он представляет собой единый способ, позволяющий для каждой задачи из искомого бесконечного множества задач за конечное число шагов найти её решение.

На понятие алгоритма можно взглянуть и с несколько иной точки зрения. Каждую задачу из бесконечного множества задач можно выразить (закодировать) некоторым словом некоторого алфавита, а решение задачи каким-то другим словом того же алфавита. В результате получим функцию, заданную на некотором подмножестве множества всех слов выбранного алфавита и принимающую значение в множестве всех слов того же алфавита. Решить какую-либо – значит найти значение этой функции на слове, кодирующем данную задачу. А иметь алгоритм для решения всех задач данного класса – значит иметь единый способ, позволяющий за конечное число шагов «вычислить» значение построенной функции для любых значений аргумента из её области определения. Таким образом, алгоритмическая проблема – по существу проблема о вычислении значений заданной функции в некотором алфавите.

Остается уточнить, что значит уметь вычислить значение функции. Это значит вычислить значение функции с помощью подходящей машины Тьюринга. Для каких же функций возможно их тьюрингово вычисление? Многочисленные исследования ученых показали, что такой класс функций чрезвычайно широк. Каждая функция, для вычисления значений которой существует какой-нибудь алгоритм, оказалась вычисляемой посредством некоторой машины Тьюринга. Это дало повод Тьюрингу высказать следующую гипотезу, называемую основной гипотезой теории алгоритмов или тезисом Тьюринга.

Тезис Тьюринга. Для нахождения значений функции, заданной в некотором алфавите, тогда и только тогда существует какой-нибудь алгоритм, когда функция является вычислимой по Тьюрингу, т.е. когда она может вычисляться на подходящей машине Тьюринга.

Это означает, что строгое математическое понятие вычислимое по Тьюрингу функции является идеальной моделью взятого из опыта понятия алгоритма. Данный тезис есть не что иное, как аксиома, постулат, о взаимосвязях нашего опыта с той математической теорией, которую мы под этот опыт хотим подвести. Конечно же, данный тезис в принципе не может быть доказан методами математики, потому что он не имеет внутри математического характера (одна сторона в тезисе – понятие алгоритма – не является точным математическим понятием). Он выдвинут, исходя из опыта, и именно опыт подтверждает его состоятельность.

Впрочем, не исключается принципиальная возможность того, что тезис Тьюринга будет опровергнут. Для этого должна быть указана функция, вычисляемая с помощью какого-нибудь алгоритма, но не вычисляемая на какой-нибудь машине Тьюринга. Но такая возможность представляется маловероятной.

Подобно тезису Тьюринга в теории рекурсивной функции выдвигается соответствующая гипотеза, носящая название тезиса Чёрча.

Тезис Чёрча. Числовая функция тогда и только тогда алгоритмически вычислима, когда она рекурсивна.

И эта гипотеза не может быть доказана строго математически, она подтверждается практикой, опытом, ибо призвана увязать практику и теорию. Все рассматриваемые в математике конкретные функции, вычислимые в алгоритмическом смысле, оказались рекурсивными.

Мы познакомились с несколькими теориями, каждая из которых уточняет понятие алгоритма. Возникает вопрос, как связаны эти теории между собой. Ответ на него дает следующая теорема.

Теорема 2.1 *Следующие классы функций (заданные на натуральных числах и принимающие натуральное значение) совпадают.*

1) Класс всех функций вычислимых по Тьюрингу;

2) Класс всех рекурсивных функций.

Это уже не гипотеза и не тезис, а математическая теорема, которая строго доказывается.

Можно отметить, что существуют ещё и другие теории алгоритмов, и для всех них также доказана их равнозначность с рассматриваемыми теориями.

В 1936 году Чёрчем было доказано, что не существует алгоритма, позволяющего для каждой формулы логики предикатов определить, будет ли она выполнимой или общезначимой.

Одной из наиболее знаменитых алгоритмических проблем математики являлась 10-я проблема Гильберта, поставленная им в числе других в 1901 году на Международном математическом конгрессе. Требовалось найти алгоритм, определяющий для любого диофантова уравнения, имеет ли оно целочисленное решение.

Диофантово уравнение есть уравнение вида $F(x, y, ..., z) = 0,$ где $F(x, y, ..., z) -$ многочлен с целыми показателями степеней и целыми коэффициентами. В общем случае эта проблема долго оставалась нерешенной и только в 1970 году советский математик Ю. В. Матиясевич доказал её неразрешимость.

В заключение ещё раз отметим, что алгоритмическая неразрешимость означает лишь отсутствие единого способа для решения всех задач данной бесконечной серии, в то же время каждая индивидуальная задача вполне может быть решена индивидуальным способом. Так, например, несмотря на отсутствие единого алгоритма, определяющего имеет ли диофантово уравнение целочисленное решение, для частного случая диофантово уравнение $a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + x_0 = 0$ хорошо известно, что все его целые корни следует искать среди делителей свободного члена a_0.

2.4 Вопросы для самоконтроля

1 Что Вы понимаете под машиной Тьюринга?

2 Из каких частей состоит машина Тьюринга?

3 Дайте определение машинного слова.

4 Какая функция называется числовой?

5 Какая функция называется частично-числовой?

6 Дайте определение функции, вычислимой по Тьюрингу.

7 Какие функции называются простейшими?

8 Дайте определение операции суперпозиции.

9 Дайте определение операции примитивной рекурсии.

10 Дайте определение операции минимизации.

11 Сформулируйте тезис Черча.

12 Сформулируйте тезис Тьюринга.

13 Какая связь между функциями, вычисляемыми по Тьюрингу и частично рекурсивными функциями?

ЛИТЕРАТУРА

1 Карпов, В.Г. Математическая логика и дискретная математика [Текст]: учебное пособие для студентов университетов / В.Г.Карпов, В.А. Мощенский. – Мн.: Вышэйшая школа, 1977. – 255с.

2 Яблонский, С.В. Введение в дискретную математику [Текст] : учебное пособие для вузов по специальности «Прикладная математика» / С.В. Яблонский. – М.: Наука, 1979. – 272с.

3 Мощенский, В.А. Лекции по математической логике [Текст] : учебное пособие для студентов математических специальностей вузов / В.А. Мощенский. – Мн.: Изд. Центр БГУ, 1973. – 159с.

4 Лавров, И.А. Задачи по теории множеств, математической логике и теории алгоритмов [Текст] : учебное пособие для студентов математических специальностей вузов / И.А. Лавров, Л.Л. Максимова. – 2-е стер. – М.: Наука, 1984. – 223с.

5 Гаврилов, Г.П. Сборник задач по дискретной математике [Текст] : учебное пособие для вузов по специальности «Прикладная математика» / Г.П. Гаврилов, А.А. Сапоженко. – М. : Наука, 1984. – 223с.

Учебное издание

Ходанович Дмитрий Александрович
Семенчук Владимир Николаевич

ТЕОРИЯ АЛГОРИТМОВ И
КОНЕЧНЫХ АВТОМАТОВ

КУРС ЛЕКЦИЙ
для студентов специальности
специальности 1-31 03 01 «Математика»

Редактор В.И. Шкредова
Корректор В.В. Калугина

Подписано в печать 15.11.2011 (???). Формат 60х90 1/16. Бумага писчая №1. Гарнитура «Таймс» Усл. печ. л. ?, . Уч.-изд. ?, . Тираж 25 экз.

Отпечатано в учреждении образования «Гомельский государственный университет имени Франциска Скорины»
246019, г. Гомель, ул. Советская, 104.