МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М. В. ЛОМОНОСОВА

Механико-математический факультет

Конспект лекций О. Б. Лупанова

по курсу

"Введение в математическую логику"

Москва 2007

УДК 510.6 : 510.7 : 519.7
ББК 22

Конспект лекций *О. Б. Лупанова* по курсу "Введение в математическую логику"/Отв. ред. А. Б. Угольников. М.: Изд-во ЦПИ при механико-математическом факультете МГУ имени М. В. Ломоносова, 2007. 192 с.

Учебное пособие составлено на основе конспектов лекций академика РАН О. Б. Лупанова по курсу "Введение в математическую логику", прочитанных им на первом курсе механико-математического факультета МГУ им. М. В. Ломоносова в 1982–2006 гг. В пособии рассматриваются следующие вопросы: функции алгебры логики, функции многозначной логики, исчисление высказываний, логика и исчисление предикатов, логические сети, конечные автоматы, алгоритмы и вычислимые функции.

Для студентов и аспирантов.

Ответственный редактор: Александр Борисович Угольников

© Лупанов О. Б., 2007

СОДЕРЖАНИЕ

Логические схемы

Конечные автоматы

Алгоритмы и вычислимые функции

Исчисление высказываний

Логика предикатов

Исчисление предикатов

Предисловие

Эта книга выходит в свет, когда Олега Борисовича Лупанова (02.06.1932 – 03.05.2006) уже нет в живых.

Лекции по математической логике О. Б. Лупанов читал на первом курсе механико-математического факультета МГУ на протяжении 25 лет с 1982 по 2006 г. Все эти годы Олег Борисович был заведующим кафедрой дискретной математики и деканом механико-математического факультета МГУ. В лекциях отражена точка зрения О. Б. Лупанова — выдающегося советского математика, академика РАН, одного из основателей отечественной школы дискретной математики и математической кибернетики — на преподавание математической логики студентам механико-математического факультета. Отличительная черта лекций — тщательный отбор излагаемого материала, простой и доступный стиль изложения. Олег Борисович предполагал подготовить к изданию свои лекции и начал работать над рукописью. Этим планам не суждено было осуществиться.

Учебное пособие составлено в значительной степени на основе конспектов лекций разных лет, любезно предоставленных студентами (многие из которых в настоящее время уже являются аспирантами и сотрудниками механико-математического факультета), использовались также черновики, статьи и учебные пособия[1)] Олега Борисовича, а кроме того, электронные версии курса[2)].

Содержание курса в разные годы менялось. При подготовке настоящего издания была предпринята попытка собрать весь материал, прочитанный О. Б. Лупановым в разные годы по данному

[1)] В первую очередь использовались учебные пособия по курсу лекций О. Б. Лупанова по математической логике, прочитанных им на факультете ВМиК МГУ в 1970/71 уч. г. (см.: *Лупанов О. Б.* Лекции по математической логике, ч. 1. М.: МГУ, факультет ВМиК, 1970. 80 с.; *Лупанов О. Б.* Лекции по математической логике, ч. 2. М.: МГУ, факультет ВМиК, 1970. 27 с.), а также следующие материалы:

– *Лупанов О. Б.* Асимптотические оценки сложности управляющих систем. М.: Изд-во МГУ, 1984. 136 с.;

– *Лупанов О. Б.* О синтезе некоторых классов управляющих систем// Проблемы кибернетики. Вып. 10. М.: Физматгиз, 1963. 63-97,

[2)] См.: http://dmvn.mexmat.net.

курсу, что отчасти объясняет некоторую неоднородность лекций относительно объема представленного в них материала. Однако осуществить эти намерения удалось не в полной мере. В настоящее издание, в частности, не включен асимптотически наилучший метод синтеза схем из функциональных элементов, не вошло описание универсальной машины Тьюринга, не отражены также некоторые другие вопросы. Эти пробелы, возможно, удастся восполнить в последующих изданиях. Материал, содержащийся в данном учебном пособии, по объему соответствует примерно годовому курсу лекций.

При работе над книгой хотелось сохранить стиль изложения, присущий О. Б. Лупанову. Насколько это удалось, судить читателям. Хочется надеяться, что данное издание будет полезно как студентам и аспирантам, так и специалистам по математической логике, дискретной математике и математической кибернетике.

***

Эта книга вышла в свет в результате совместных усилий большого числа людей, в адрес которых необходимо сказать слова благодарности.

Прежде всего хочется поблагодарить Н. Б. Лупанову, которая дала согласие на издание лекций и предоставила в распоряжение автора этих строк черновики и рукописи Олега Борисовича для подготовки настоящего издания, Р. М. Колпакова за подробный (написанный в аспирантские годы) конспект лекций 1992/93 уч. г., а также всех студентов, аспирантов и сотрудников механико-математического факультета за предоставленные конспекты.

Особую благодарность заслуживает О. М. Касим-Заде, инициатива которого во многом способствовала тому, что данное пособие было написано и опубликовано.

Отдельную благодарность заслуживают П. А. Бородин и Ю. В. Бородина за кропотливую работу по подготовке электронной версии текста, Н. А. Леонтьева за тщательное редактирование рукописи, В. М. Староверов и О. С. Дудакова за деятельное участие в подготовке окончательной версии оригинала-макета.

Хочется выразить также искреннюю признательность сотрудникам кафедры дискретной математики за плодотворные обсуждения, коллегам и друзьям за моральную поддержку.

***

В эти дни исполняется 75 лет со дня рождения Олега Борисовича Лупанова, памяти которого посвящается настоящая книга.

2 июня 2007 г. А. Б. Угольников

ФУНКЦИИ АЛГЕБРЫ ЛОГИКИ

Лекция № 1

Будем рассматривать функции $f(x_1, \ldots, x_n)$, определенные на множестве наборов $(\sigma_1, \ldots, \sigma_n)$ из нулей и единиц и принимающие на каждом из этих наборов значения 0 или 1. Такие функции называются *булевыми функциями* или *функциями алгебры логики*. Множество всех таких функций обозначается через P_2. Так как наборов $(\sigma_1, \ldots, \sigma_n)$ длины n из нулей и единиц конечное число (именно 2^n штук!), то каждая функция может быть полностью задана таблицей (табл. 1).

Таблица 1

x_1	x_2	$\ldots$	x_{n-1}	x_n	$f(x_1, x_2, \ldots, x_{n_1}, x_n)$
0	0	$\ldots$	0	0	$f(0, 0, \ldots, 0, 0)$
0	0	$\ldots$	0	1	$f(0, 0, \ldots, 0, 1)$
		$\ldots$			$\ldots$
σ_1	σ_2	$\ldots$	σ_{n-1}	σ_n	$f(\sigma_1, \sigma_2, \ldots, \sigma_{n-1}, \sigma_n)$
		$\ldots$			$\ldots$
1	1	$\ldots$	1	1	$f(1, 1, \ldots, 1, 1)$

В левой части табл. 1 выписаны все наборы значений переменных, в правой части — соответствующие им значения функций. На каждом из 2^n наборов функция $f(x_1, \ldots, x_n)$ может иметь любое из двух значений. Отсюда следует

Теорема. *Число функций алгебры логики от n переменных $x_1, x_2, \ldots, x_n$ равно 2^{2^n}.*

Это число обозначается через $p_2(n)$. Из теоремы следует, что, с одной стороны, число функций от фиксированного (конечного) множества переменных конечно, а с другой — это число "невероятно быстро" растет с ростом числа переменных. Действительно, $p_2(n+1) = 2^{2^{n+1}} = 2^{2^n \cdot 2} = (2^{2^n})^2 = p_2^2(n)$; т. е. при увеличении числа аргументов на 1 число функций алгебры логики возводится в квадрат. Например,

$$2^{2^0} = 2, \quad 2^{2^1} = 4, \quad 2^{2^2} = 16, \quad 2^{2^3} = 256,$$
$$2^{2^4} = 65\,536, \quad 2^{2^5} = 4\,294\,967\,296.$$

Заметим, что с теоретико-множественной точки зрения функция алгебры логики $f(x_1, \ldots, x_n)$ — это не просто отображение

множества всех наборов длины n из нулей и единиц в множество $\{0,1\}$, а совокупность такого отображения и упорядоченного набора $(x_1,\dots,x_n)$ переменных. В этом смысле функции $f(x_1,x_2,x_3,\dots,x_n)$ и $f(x_2,x_1,x_3,\dots,x_n)$, вообще говоря, различны, хотя и определяются одним и тем же отображением $\{0,1\}^n \to \{0,1\}$.

В дальнейшем важную роль будут играть некоторые функции одного и двух аргументов, которые являются в определенном смысле аналогами "элементарных функций" в арифметике, алгебре и математическом анализе. Рассмотрим эти функции подробно.

При $n=1$ будет всего 4 функции, указанные в табл. 2, где 0 — *константа* 0, x — *тождественная функция*, $\overline{x}$ — *отрицание* x, 1 — *константа* 1; функция $\overline{x}$ играет особо важную роль в математической логике.

Таблица 2

x	0	x	$\overline{x}$	1
0	0	0	1	1
1	0	1	0	1

При $n=2$ будет уже 16 функций. Некоторые из них указаны в табл. 3.

Таблица 3

x_1	x_2	$x_1\&x_2$	$x_1 \vee x_2$	$x_1 \oplus x_2$	$x_1 \to x_2$	x_1/x_2
0	0	0	0	0	1	1
0	1	0	1	1	1	1
1	0	0	1	1	0	1
1	1	1	1	0	1	0

Функция $x_1\&x_2$ называется *конъюнкцией* x_1 и x_2 или *логическим умножением* (И), она обозначается также $x_1 \cdot x_2$ или x_1x_2. Функция $x_1 \vee x_2$ называется *дизъюнкцией* x_1 и x_2 или *логическим сложением* (ИЛИ); $x_1 \oplus x_2$ называется *суммой* x_1 и x_2 *по модулю* 2; $x_1 \to x_2$ называется *импликацией* x_1 и x_2 (эта функция имеет важное значение в математической логике); x_1/x_2 — это *штрих Шеффера*.

Переменная x_i функции $f(x_1,\dots,x_n)$ называется *существенной*, если существуют такие два набора $(\alpha_1,\dots,\alpha_{i-1},0,\alpha_{i+1},\dots,\alpha_n)$ и $(\alpha_1,\dots,\alpha_{i-1},1,\alpha_{i+1},\dots,\alpha_n)$, различающиеся только в i-й компоненте, что

$$f(\alpha_1,\dots,\alpha_{i-1},0,\alpha_{i+1},\dots,\alpha_n) \neq f(\alpha_1,\dots,\alpha_{i-1},1,\alpha_{i+1},\dots,\alpha_n).$$

В этом случае говорят, что функция $f(x_1,\dots,x_n)$ *существенно зависит* от переменной x_i. Переменная x_i, не являющаяся существенной, называется *несущественной* или *фиктивной* переменной функции $f(x_1,\dots,x_n)$; в этом случае говорят, что функция $f(x_1,\dots,x_n)$ не зависит существенно от переменной x_i.

Пример. Функция $f(x_1,x_2)=x_1x_2$ существенно зависит от переменной x_1, так как $f(0,1)\neq f(1,1)$. Она также существенно зависит от переменной x_2, так как $f(1,0)\neq f(1,1)$.

Аналогично показывается, что все функции, приведенные в табл. 3, существенно зависят от обеих переменных. Очевидно, что константы 0 и 1 не имеют существенных переменных.

Заметим, что значение функции полностью определяется набором значений ее существенных переменных.

Пусть функция $f(x_1,\dots,x_n)$ несущественно зависит от переменной x_i. Тогда если $f(\alpha_1,\dots,\alpha_{i-1},0,\alpha_{i+1},\dots,\alpha_n)=\beta$, то выполняется и равенство $f(\alpha_1,\dots,\alpha_{i-1},1,\alpha_{i+1},\dots,\alpha_n)=\beta$. Поэтому эта функция может быть задана в виде табл. 4.

Таблица 4

x_1	…	x_{i-1}	x_i	x_{i+1}	…	x_n	$f(x_1,\dots,x_n)$
			…				…
α_1	…	α_{i-1}	0	α_{i+1}	…	α_n	β
			…				…
α_1	…	α_{i-1}	1	α_{i+1}	…	α_n	β
			…				…

Вычеркнем в этой таблице i-й столбец и все наборы, у которых i-я компонента равна 1. Получим новую функцию $g(x_1,\dots,x_{i-1},x_{i+1},\dots,x_n)$, такую, что

$$g(\alpha_1,\dots,\alpha_{i-1},\alpha_{i+1},\dots,\alpha_n) = f(\alpha_1,\dots,\alpha_{i-1},0,\alpha_{i+1},\dots,\alpha_n)$$

для любых $\alpha_1,\dots,\alpha_{i-1},\alpha_{i+1},\dots,\alpha_n$ из множества $\{0,1\}$. Эта функция представима в виде табл. 5.

Таблица 5

x_1	...	x_{i-1}	x_{i+1}	...	x_n	$g(x_1,\dots,x_{i-1},x_{i+1},\dots,x_n)$
		...				...
α_1	...	α_{i-1}	α_{i+1}	...	α_n	$f(\alpha_1,\dots,\alpha_{i-1},0,\alpha_{i+1},\dots,\alpha_n)$
		...				...

Будем говорить, что функция g получилась из функции f *удалением несущественной переменной* x_i.

Аналогично можно ввести и, так сказать, обратную операцию — *добавление несущественной переменной* (оговорка "так сказать" имеет тот смысл, что эта операция неоднозначна). Пусть дана функция $f(x_1,\dots,x_n)$. Построим новую функцию $h(x_1,\dots,x_n,x_{n+1})$. Значение этой функции на любом наборе $(\alpha_1,\dots,\alpha_n,\alpha_{n+1})$ определяется равенством $h(\alpha_1,\dots,\alpha_n,\alpha_{n+1}) = f(\alpha_1,\dots,\alpha_n)$. Тогда x_{n+1} будет несущественной переменной функции h, поскольку $h(\alpha_1,\dots,\alpha_n,0) = f(\alpha_1,\dots,\alpha_n) = h(\alpha_1,\dots,\alpha_n,1)$. Будем говорить, что функция h получилась из функции f *добавлением несущественной переменной* x_{n+1}. Неоднозначность операции добавления несущественной переменной обусловлена тем, что в качестве таких переменных могут использоваться различные переменные.

Мы не будем различать функции, получающиеся друг из друга добавлением несущественных переменных. Две функции называются *равными*, если одна из них получается из другой в результате добавления и (или) удаления несущественных переменных.

Пример. Пусть дана функция $h_1(x_1,x_2)$ (см. табл. 6).

Таблица 6

x_1	x_2	$h_1(x_1,x_2)$
0	0	0
0	1	1
1	0	0
1	1	1

→

x_2	$g(x_2)$
0	0
1	1

→

x_2	x_3	$h_2(x_2,x_3)$
0	0	0
0	1	0
1	0	1
1	1	1

Эта функция несущественно зависит от переменной x_1. Удалим эту переменную. Получим функцию $g(x_2)$. Добавим затем несущественную переменную x_3. Получим функцию $h_2(x_2,x_3)$. Функции h_1 и h_2 равны (хотя таблицы у них разные!).

Формулы. Пусть дано некоторое (конечное или счетное) множество функций

$$F = \{f_1(x_1,\dots,x_{n_1}), f_2(x_1,\dots,x_{n_2}),\dots, f_s(x_1,\dots,x_{n_s}),\dots\}.$$

Введем понятие формулы над множеством F. Это понятие определяется индуктивно.

1. Выражения $f_i(x_1,\dots,x_{n_i})$ (т. е. знак функции, левая скобка, переменные функции f_i в их порядке, правая скобка) являются формулами над F.
2. Если $A_1,\dots,A_{n_i}$ — либо переменные, либо формулы над F, то $f_i(A_1,\dots,A_{n_i})$ — формула над F; выражения $A_1,\dots,A_{n_i}$ (отличные от символов переменных) называются подформулами формулы F.

Заметим, что при образовании новых формул вместо переменных исходных функций можно подставлять как формулы, так и переменные. Переменная, вообще говоря, не является формулой; переменная является формулой, если она входит в систему F (и обозначается тем же символом).

Пример. Пусть $F = \{\varphi(x_1,x_2)\}$. Тогда выражения $\varphi(x_1,x_2)$, $\varphi(x_1,x_1)$, $\varphi(x_2,\varphi(x_3,x_4))$ будут формулами над F, а $\varphi(x_1,x_2,x_3)$ не будет (φ имеет две переменные!).

Каждой формуле сопоставим некоторую функцию алгебры логики. Пусть дана формула Φ над множеством функций $F = \{f_1(x_1,\dots,x_{n_1}),\dots, f_s(x_1,\dots,x_{n_s}),\dots\}$, содержащая переменные $x_1,\dots,x_n$ и не содержащая никаких других переменных (т. е. $\{x_1,\dots,x_n\}$ — множество всех ее переменных). И пусть $R = (\alpha_1,\dots,\alpha_n)$ — некоторый набор значений переменных. Определим значение формулы Φ на наборе переменных R (обозначение $\Phi|_R$) индуктивно.

1. Значения переменной x_i на наборе R равно $x_i|_R = \alpha_i$.
2. Пусть уже определены значения $A_1|_R,\dots,A_{n_i}|_R$. Тогда $f_i(A_1,\dots,A_{n_i})|_R = f_i(A_1|_R,\dots,A_{n_i}|_R)$.

Так как мы можем определить значение формулы Φ на любом наборе переменных, то тем самым мы сопоставим этой формуле некоторую функцию $f(x_1,\dots,x_n)$. Про функцию, сопоставленную указанным выше способом формуле, говорят, что она *реализуется* или *выражается* этой формулой. Таким образом, каждая формула выражает какую-то функцию алгебры логики.

Формулы, реализующие равные функции, называются *эквивалентными*.

Рассмотрим множество $F = \{x_1\&x_2,\ x_1 \vee x_2,\ x_1 \oplus x_2,\ \overline{x},\ x,\ 0,\ 1\}$. Отметим несколько важных примеров эквивалентных формул над F. При этом для сокращения записи введем некоторые соглашения:

а) будем опускать внешние скобки;

б) не будем заключать в скобки переменные и константы.

1. Коммутативность операций $\&$, $\vee$, $\oplus$:

$$x_1\&x_2 = x_2\&x_1,$$
$$x_1 \vee x_2 = x_2 \vee x_1,$$
$$x_1 \oplus x_2 = x_2 \oplus x_1.$$

2. Ассоциативность операций $\&$, $\vee$, $\oplus$:

$$x_1\&(x_2\&x_3) = (x_1\&x_2)\&x_3,$$
$$x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3,$$
$$x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3.$$

3. Дистрибутивность:

$$(x_1 \vee x_2)\&x_3 = (x_1\&x_3) \vee (x_2\&x_3),$$
$$(x_1 \oplus x_2)\&x_3 = (x_1\&x_3) \oplus (x_2\&x_3),$$
$$(x_1\&x_2) \vee x_3 = (x_1 \vee x_3)\&(x_2 \vee x_3).$$

4. Закон поглощения:

$$x_1\&(x_1 \vee x_2) = x_1,$$
$$x_1 \vee (x_1\&x_2) = x_1.$$

5. Идемпотентность конъюнкции и дизъюнкции:

$$x\&x = x, \quad x \vee x = x.$$

6. Перенос отрицания через конъюнкцию и дизъюнкцию:

$$\overline{(x_1\&x_2)} = \overline{x}_1 \vee \overline{x}_2,$$
$$\overline{(x_1 \vee x_2)} = \overline{x}_1\&\overline{x}_2.$$

7. "Снятие" двойного отрицания:

$$\overline{\overline{x}} = x.$$

8. Некоторые эквивалентности с использованием отрицания и суммы по модулю 2:

$$x\&\overline{x} = 0, \quad x \vee \overline{x} = 1,$$
$$x \oplus \overline{x} = 1, \quad x \oplus x = 0.$$

9. Операции с константами:

$$x\&1 = x, \quad x\&0 = 0,$$
$$x \vee 1 = 1, \quad x \vee 0 = x,$$
$$x \oplus 1 = \overline{x}, \quad x \oplus 0 = x.$$

Эти равенства легко проверяются путем вычисления значений функций в левой и правой части равенств на каждом наборе значений переменных. Проверим, например, первое равенство в п. 3.

x_1	x_2	x_3	$x_1 \vee x_2$	$(x_1 \vee x_2)x_3$	x_1x_3	x_2x_3	$x_1x_3 \vee x_2x_3$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	0	1	1
1	0	0	1	0	0	0	0
1	0	1	1	1	1	0	1
1	1	0	1	0	0	0	0
1	1	1	1	1	1	1	1

Из этих "элементарных" правил легко вывести некоторые производные правила (или прямо доказать их справедливость).

$1'$, $2'$. В дизъюнкции из нескольких членов можно произвольным образом их переставлять и произвольным образом расставлять скобки. Аналогично для конъюнкции и суммы по модулю 2. Например,

$$((x_1 \vee x_2) \vee (x_3 \vee x_4)) \vee x_5 = (x_1 \vee (x_3 \vee x_5)) \vee (x_4 \vee x_2).$$

Это правило позволяет ввести дальнейшие соглашения, упрощающие вид формул:

в) в формулах, получающихся многократным применением операции $\vee$ к более простым формулам, будем опускать скобки.

Аналогично для операций конъюнкции и суммы по модулю 2. Например, вместо

$$((A_1 \vee A_2) \vee A_3) \vee (A_4 \vee A_5)$$

будем писать

$$A_1 \vee A_2 \vee A_3 \vee A_4 \vee A_5.$$

Введем некоторые соглашения для записи формул:

$$\&_{i=1}^{n} A_i = A_1 \& A_2 \& \ldots \& A_n,$$
$$\bigvee_{i=1}^{n} A_i = A_1 \vee A_2 \vee \cdots \vee A_n,$$
$$\sum_{i=1}^{n} A_i = A_1 \oplus A_2 \oplus \cdots \oplus A_n.$$

Далее нам часто будет удобно нумеровать формулы не натуральными числами, а наборами из нулей и единиц. Например,

$$A_{000} \vee A_{001} \vee A_{010} \vee A_{011} \vee A_{100} \vee A_{101} \vee A_{110} \vee A_{111}.$$

В этом случае будем использовать сокращение

$$\bigvee_{\sigma_1=0}^{1} \bigvee_{\sigma_2=0}^{1} \bigvee_{\sigma_3=0}^{1} A_{\sigma_1\sigma_2\sigma_3}$$

или короче

$$\bigvee_{(\sigma_1\sigma_2\sigma_3)} A_{\sigma_1\sigma_2\sigma_3}.$$

Введем функцию

$$x^\sigma = \begin{cases} x, \text{ если } \sigma = 1; \\ \overline{x}, \text{ если } \sigma = 0 \end{cases}$$

(то, что $x^1 = x$, привычно; обычно $x^0 = 1$, но в алгебре логике удобно считать, что $x^0 = \overline{x}$). Эта функция обладает следующим

свойством: x^σ принимает значение 1 тогда и только тогда, когда x принимает значение σ. Отсюда следует, что конъюнкция

$$x_1^{\sigma_1} x_2^{\sigma_2} \ldots x_n^{\sigma_n}$$

на наборе $(\sigma_1, \sigma_2, \ldots, \sigma_n)$ принимает значение 1, а на всяком другом наборе — значение 0.

Теорема (о разложении функции по множеству переменных)**.** *Пусть даны функция $f(x_1, x_2, \ldots, x_n)$ и число k, $1 \leqslant k \leqslant n$. Тогда функцию f можно представить в следующей форме:*

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{(\sigma_1,\ldots,\sigma_k)} x_1^{\sigma_1} \ldots x_k^{\sigma_k} f(\sigma_1, \ldots, \sigma_k, x_{k+1}, \ldots, x_n),$$

где дизъюнкция берется по всевозможным наборам значений переменных $(x_1, \ldots, x_k)$.

Доказательство. Возьмем произвольный набор $(\alpha_1, \ldots, \alpha_n)$. Найдем значения формулы на этом наборе. Если $(\sigma_1, \ldots, \sigma_k) = (\alpha_1, \ldots, \alpha_k)$, то выполняется равенство

$$\alpha_1^{\sigma_1} \ldots \alpha_k^{\sigma_k} f(\sigma_1, \ldots, \sigma_k, \alpha_{k+1}, \ldots, \alpha_n) = f(\alpha_1, \ldots, \alpha_n).$$

Если $(\sigma_1, \ldots, \sigma_k) \neq (\alpha_1, \ldots, \alpha_k)$, то

$$\alpha_1^{\sigma_1} \ldots \alpha_k^{\sigma_k} f(\sigma_1, \ldots, \sigma_k, \alpha_{k+1}, \ldots, \alpha_n) = 0.$$

То есть

$$\bigvee_{(\sigma_1,\ldots,\sigma_k)} \alpha_1^{\sigma_1} \ldots \alpha_k^{\sigma_k} f(\sigma_1, \ldots, \sigma_k, \alpha_{k+1}, \ldots, \alpha_n) = f(\alpha_1, \ldots, \alpha_n).$$

Отметим, что мы получили представление функции f в виде формулы над множеством

$$\{\vee, \&, \overline{\ }, f(0, \ldots, 0, x_{k+1}, \ldots, x_n),$$
$$f(0, \ldots, 0, 1, x_{k+1}, \ldots, x_n), \ldots, f(1, \ldots, 1, x_{k+1}, \ldots, x_n)\}.$$

Рассмотрим отдельно два частных случая: $k = 1$ и $k = n$.

1) $k = 1$. В этом случае мы получим разложение функции f по первой переменной

$$f(x_1, \dots, x_n) = x_1^0 f(0, x_2, \dots, x_n) \vee x_1^1 f(1, x_2, \dots, x_n) =$$

$$= x_1 f(1, x_2, \dots, x_n) \vee \overline{x}_1 f(0, x_2, \dots, x_n).$$

2) $k = n$. Тогда

$$f(x_1, \dots, x_n) = \bigvee_{(\sigma_1, \dots, \sigma_n)} x_1^{\sigma_1} \dots x_n^{\sigma_n} f(\sigma_1, \dots, \sigma_n).$$

Так как $f(\sigma_1, \dots, \sigma_n)$ — это либо 0, либо 1, то в этой дизъюнкции нам достаточно оставить лишь такие наборы, для которых выполняется равенство $f(\sigma_1, \dots, \sigma_n) = 1$. Получим

$$f(x_1, \dots, x_n) = \bigvee_{(\sigma_1, \dots, \sigma_n) | f(\sigma_1, \dots, \sigma_n) = 1} x_1^{\sigma_1} \dots x_k^{\sigma_n}.$$

Это выражение будет формулой над множеством $\{\vee, \&, \overline{\ }\}$. Такое представление функции f называется *совершенной дизъюнктивной нормальной формой* (*СДНФ*).

Таким образом, каждую функцию алгебры логики можно выразить в виде формулы над множеством $\{\vee, \&, \overline{\ }\}$.

Пример.

$$x \to y = \overline{x}\&\overline{y} \vee \overline{x}\&y \vee x\&y,$$

$$x \oplus y = \overline{x}\&y \vee x\&\overline{y}.$$

Отметим, что функция, равная константе 0 (т. е. принимающая значение 0 на любом наборе значений переменных), не имеет представления в виде СДНФ.

Лекция № 2

Система булевых функций F называется *полной*, если любая функция алгебры логики выражается формулой над F. В конце прошлой лекции мы доказали, что $F = \{\vee, \&, \overline{\ }\}$ является полной системой. В дальнейшем будет установлен критерий полноты систем функций из P_2. Но предварительно мы приведем еще некоторые примеры полных систем. Полнота систем функций может устанавливаться с использованием следующего простого утверждения (достаточного условия полноты системы булевых функций).

Теорема. *Пусть даны системы булевых функций F и G, такие, что F полная и любая функция из F выражается формулой над G. Тогда G — полная система.*

Доказательство. Пусть $F = \{f_1, f_2, \dots, f_s, \dots\}$ — полная система, а $\Phi_1, \Phi_2, \dots, \Phi_s, \dots$ — формулы над G, выражающие функции $f_1, f_2, \dots, f_s, \dots$ соответственно. Возьмем произвольную функцию f из P_2. Так как F — полная система, то существует формула Φ над F, выражающая эту функцию. Заменим в формуле Φ каждую из функций f_i на соответствующую ей подформулу Φ_i. В результате мы получим формулу над системой G, реализующую функцию f. Следовательно, любая функция из P_2 выражается формулой над G. Поэтому система G полная.

Теорема. *Из любой полной системы можно выделить конечную полную подсистему.*

Доказательство. Пусть F — полная (возможно, бесконечная) система. Тогда существуют формулы $\Phi_\&$, $\Phi_\vee$ и $\Phi_{\overline{\ }}$ над F, выражающие функции $\&$, $\vee$ и $\overline{\ }$ соответственно. В каждую из этих формул входит конечное число функций из F. Возьмем все эти функции. Получим систему F_1, которая является подсистемой системы F и состоит из конечного числа функций. Так как функции $\&$, $\vee$, $\overline{\ }$ выражаются формулами над F_1, а $\{\&, \vee, \overline{\ }\}$ — полная система, то в силу достаточного условия полноты F_1 также является полной (конечной) системой.

Используя достаточное условие полноты, построим еще несколько полных систем.

1. Система $F = \{\vee, \overline{\ }\}$ полная. Действительно, очевидно, что конъюнкция и отрицание выражаются формулами над F. Так как $x_1 \vee x_2 = (\overline{\overline{x}_1 \& \overline{x}_2})$, то и дизъюнкция выражается формулой над F. Следовательно, эта система является полной.

2. Система $\{\&, \overline{\ }\}$ полная. Доказывается аналогичным способом, поскольку $x_1 x_2 = (\overline{\overline{x}_1 \vee \overline{x}_2})$.

3. Система $F = \{\&, \oplus, 1\}$ полная. Действительно, конъюнкция и отрицание ($\overline{x} = x + 1$) выражаются формулами над F. Поэтому, так как $\{\&, \overline{\ }\}$ — полная система, то и F — полная система.

4. Система $F = \{/\}$ (где "/" — штрих Шеффера) полная. Выразим отрицание и конъюнкцию через штрих Шеффера:

$$\overline{x} = x/x, \quad x_1 x_2 = \overline{x_1/x_2} = (x_1/x_2)/(x_1/x_2).$$

Следовательно, F является полной системой (состоящей из одной функции).

Существует еще одна функция (двух переменных), образующая полную систему, — это функция $x_1 \downarrow x_2 = \overline{x_1 \vee x_2}$ (стрелка Пирса). Доказательство аналогично.

Итак, мы доказали, что следующие системы являются полными:

$$\{\&, \vee, \overline{\ }\}, \quad \{\&, \overline{\ }\}, \quad \{\vee, \overline{\ }\}, \quad \{\&, \oplus, 1\}, \quad \{/\}, \quad \{\downarrow\}.$$

Полиномы Жегалкина. Рассмотрим конъюнкции вида $x_{i_1} \dots x_{i_k}$, где все $i_1, \dots, i_k$ различны, $k \geqslant 1$. При $k = 1$ получаем конъюнкции длины 1, т. е. переменные. Будем рассматривать также константу 1 и называть ее конъюнкцией длины 0 (от пустого множества переменных).

Полиномом Жегалкина называется сумма по модулю 2 попарно различных конъюнкций. *Пустой* полином Жегалкина (т. е. не содержащий конъюнкций) по определению выражает константу 0.

Таким образом, полином Жегалкина — это выражение вида

$$\sum_{\{i_1, \dots, i_k\}} x_{i_1} \dots x_{i_k}$$

(где суммирование происходит по различным подмножествам множества $\{1, 2, \dots, n\}$).

Теорема (И.И. Жегалкин). *Любая функция алгебры логики представима в виде полинома Жегалкина, причем это представление единственно с точностью до перестановки слагаемых и перестановки множителей в слагаемых.*

Доказательство. Докажем сначала представимость любой булевой функции в виде полинома. Так как $\{\&, \oplus, 1\}$ — полная система, то каждая функция $f(x_1, \dots, x_n)$ представима формулой Ф над $\{\&, \oplus, 1\}$. Преобразуем эту формулу в полином Жегалкина следующим образом.

1. Приведение формулы Ф к виду

$$A_1 \oplus \dots \oplus A_l,$$

где $A_1, \dots, A_l$ — формулы над $\{\&, 1\}$. Этого можно добиться, раскрывая все скобки и применяя каждый раз дистрибутивный закон $(A_1 \oplus A_2)A_3 = A_1 A_2 \oplus A_2 A_3$, когда какая-нибудь сумма умножается на другую формулу.

2. Удаление повторяющихся переменных в произведениях. Так как $xx = x$, то в каждом произведении можно оставить только по одному экземпляру каждой встречающейся в нем переменной.

3. Удаление лишних единиц. Так как $x\&1 = x$, то мы можем удалить все 1 во всех произведениях, содержащих как переменные, так и 1. Если же какое-нибудь произведение имеет вид $1\& \dots \&1$ (т. е. не содержит переменных), то мы преобразуем его к виду 1.

4. Приведение подобных. Так как $x \oplus x = 0$, то при наличии пары одинаковых слагаемых (с точностью до перестановки множителей) оба этих слагаемых можно удалить.

В итоге мы получим полином Жегалкина. При этом в случае, когда при приведении подобных пропадут все слагаемые, этот полином будет пустым.

Докажем теперь единственность. Так как любое произведение либо содержит, либо не содержит каждую из переменных $x_1, \dots, x_n$, то всего различных произведений из переменных $x_1, \dots, x_n$ будет 2^n. При этом вместо пустого произведения (не содержащего ни одной из переменных) мы берем константу 1. Далее, так как любой полином либо содержит, либо не содержит каждое из 2^n произведений, то всего различных полиномов Жегалкина (включая пустой) будет ровно 2^{2^n}, т. е. столько же, сколько и всех функций в P_2 от переменных $x_1, \dots, x_n$. Поэтому если какая-нибудь функция представима в виде двух различных полиномов Жегалкина, то найдется функция, которая не будет представима в виде полинома (так как каждый полином выражает ровно одну функцию), что невозможно. Теорема доказана.

Рассмотрим полиномы Жегалкина, состоящие из произведе-

ний, длина которых не превышает 1. Каждый такой полином реализует некоторую функцию вида $x_{i_1} \oplus x_{i_2} \oplus \dots \oplus x_{i_k} \oplus c$, где $c \in \{0,1\}$, $k \geqslant 0$. Такие функции называются *линейными*. Множество всех линейных булевых функций обозначается через L.

Пусть дана система булевых функций F. *Замыканием* F называется множество $[F]$, состоящее из всех функций, выражаемых формулами над F.

Примеры.

1. Пусть $F = \{x_1 \oplus x_2\}$. Тогда множество $[F]$ содержит суммы любого числа переменных (в том числе константу 0 и сами переменные, так как $x \oplus x = 0$ и $x \oplus x \oplus x = x$). То есть замыкание множества F — это множество всех однородных линейных функций (т. е. таких, у которых свободный член $c = 0$).

2. Пусть $F = \{x_1 \oplus x_2, 1\}$. Тогда замыкание F — это просто множество всех линейных функций, т. е. $L = [\{x_1 \oplus x_2, 1\}]$.

Отметим некоторые свойства операции замыкания.

1. $F \subseteq [F]$.
2. Если $F_1 \subseteq F_2$, то $[F_1] \subseteq [F_2]$.
3. $[F_1 \cup F_2] \supseteq [F_1] \cup [F_2]$.
4. $[[F]] = [F]$.
5. Если F — полная система, то $[F] = P_2$.

Множество F называется *замкнутым*, если $F = [F]$. Замкнутые множества функций называют также замкнутыми классами.

Пример. Множество L является замкнутым. Это множество является нетривиальным, т. е. $L \neq \emptyset$ и $L \neq P_2$.

Докажем следующую лемму.

Лемма (о нелинейной функции). *Из любой нелинейной функции, подставляя вместо некоторых переменных константы* 0 *и* 1 *и, может быть, отрицание переменных, а также, может быть, навешивая отрицание на функцию, можно получить конъюнкцию двух переменных.*

Доказательство. Пусть $f(x_1, \dots, x_n) \not\subseteq L$. Рассмотрим представление f в виде полинома Жегалкина. Тогда в этом полиноме есть произведение, длина которого больше 1 (нелинейное произведение). Возьмем самое короткое из них. Пусть оно имеет вид $x_1 \dots x_p$, $p \geqslant 2$. Тогда $f(x_1, \dots, x_n) = x_1 \dots x_p \oplus A_1 \oplus \dots \oplus A_l$. Каждое другое нелинейное произведение содержит хотя бы одну переменную, отличную от переменных $x_1, \dots, x_p$. Подставим константу 0 вместо всех переменных $x_{p+1}, \dots, x_n$. Тогда все остальные конъюнкции (из $A_1, \dots, A_l$) обратятся в нуль. Поэтому

$$f(x_1, \dots, x_p, 0, \dots, 0) = x_1 \dots x_p \oplus l(x_1, \dots, x_p),$$

где $l(x_1, \dots, x_p)$ — некоторая линейная функция от переменных $x_1, \dots, x_p$.

Далее, оставим две первые переменные без изменения, а вместо остальных (если они есть) подставим 1. Получим функцию

$$f(x_1, x_2, 1, \dots, 1, 0, \dots, 0) = x_1 x_2 \oplus l(x_1, x_2) = x_1 x_2 \oplus a x_1 \oplus b x_2 \oplus c,$$

где $a, b, c \in \{0,1\}$. Обозначим эту функцию через $g(x_1, x_2)$.

Поставим в функцию g вместо переменных x_1 и x_2 функции $x_1 \oplus b$ и $x_2 \oplus a$ соответственно и прибавим ко всей функции константу $ab \oplus c$ (т. е. либо ничего не изменим, либо навесим отрицание). Получим

$$g(x_1 \oplus b, x_2 \oplus a) \oplus (ab \oplus c) =$$
$$= (x_1 \oplus b)(x_2 \oplus a) \oplus a(x_1 \oplus b) \oplus b(x_2 \oplus a) \oplus c \oplus (ab \oplus c) =$$
$$= x_1 x_2 \oplus x_1 a \oplus b x_2 \oplus ba \oplus a x_1 \oplus ab \oplus b x_2 \oplus ba \oplus c \oplus (ab \oplus c) = x_1 x_2.$$

Следствие. *Если* $f \notin L$, *то* $xy \in [\{f, 0, 1, \bar{}\}]$.

Рассмотрим также еще некоторые замкнутые классы.

Обозначим через T_0 множество всех булевых функций $f(x_1, \dots, x_n)$, таких, что выполняется равенство $f(0, \dots, 0) = 0$, $n \geqslant 1$. Это множество нетривиально (т. е. непусто и не совпадает с P_2). Покажем, что T_0 — замкнутый класс. Очевидно, что тождественная функция принадлежит этому классу. Поэтому достаточно показать, что если $f_0(x_1, \dots, x_n) \in T_0$ и $f_1, \dots, f_n \in T_0$, то и функция $f = f_0(f_1, \dots, f_n)$ принадлежит T_0. Действительно, так как $f_i(0, \dots, 0) = 0$ при всех $i = 0, 1, \dots, n$, то

$$f(0, \dots, 0) = f_0(f_1(0, \dots, 0), \dots, f_n(0, \dots, 0)) = f_0(0, \dots, 0) = 0.$$

Следовательно, T_0 — замкнутый класс.

Обозначим через T_1 множество всех булевых функций $f(x_1, \dots, x_n)$, таких, что $f(1, \dots, 1) = 1$, $n \geqslant 1$. Этот класс также нетривиален и замкнут (доказательство аналогично).

Пусть $f(x_1,\dots,x_n)$ — произвольная функция алгебры логики. Функция $f^*(x_1,\dots,x_n)=\overline{f}(\overline{x}_1,\dots,\overline{x}_n)$ называется *двойственной* к функции f.

Примеры.

$$(x)^*=\overline{\overline{x}}=x,\quad (\overline{x})^*=\overline{\overline{x}}=\overline{x},$$
$$0^*=\overline{0}=1,\quad 1^*=\overline{1}=0,$$
$$(x_1\&x_2)^*=(\overline{\overline{x}_1\&\overline{x}_2})=x_1\vee x_2,$$
$$(x_1\vee x_2)^*=x_1x_2.$$

Как видно из этих примеров, для некоторых функций f выполняется равенство $f^*=f$. Такие функции называются *самодвойственными*. Множество всех самодвойственных функций алгебры логики обозначается через S. Очевидно, что множество S нетривиально. Докажем, что оно замкнуто.

Из приведенных выше примеров видно, что тождественная функция принадлежит множеству S. Поэтому нам и в этом случае достаточно доказать, что если $f_0(x_1,\dots,x_n)\in S$ и $f_1,\dots,f_n\in S$, то и функция $f=f_0(f_1,\dots,f_n)$ принадлежит S.

Будем считать, что у всех функций f_i, $i=1,\dots,n$, одинаковый набор переменных $y_1,\dots,y_m$ (так как в противном случае недостающие переменные можно добавить, применяя операцию добавления несущественных переменных). То есть

$$f(y_1,\dots,y_m)=f_0(f_1(y_1,\dots,y_m),\dots,f_n(y_1,\dots,y_m)).$$

Тогда

$$\begin{aligned}f^*(y_1,\dots,y_m)&=\overline{f_0}(f_1(\overline{y}_1,\dots,\overline{y}_m),\dots,f_n(\overline{y}_1,\dots,\overline{y}_m))=\\&=\overline{f_0}(\overline{f_1^*}(y_1,\dots,y_m),\dots,\overline{f_n^*}(y_1,\dots,y_m))=\\&=f_0^*(f_1^*,\dots,f_n^*)=f(y_1,\dots,y_m),\end{aligned}$$

так как $f_0,f_1,\dots,f_n\in S$.

Отметим, что для любой функции $f(x_1,\dots,x_n)$ равенство $\overline{f}(\overline{x}_1,\dots,\overline{x}_n)=f(x_1,\dots,x_n)$ выполняется тогда и только тогда, когда выполняется равенство $\overline{\overline{f}}(\overline{x}_1,\dots,\overline{x}_n)=\overline{f}(x_1,\dots,x_n)$, т. е. когда $f(\overline{x}_1,\dots,\overline{x}_n)=\overline{f}(x_1,\dots,x_n)$. Поэтому каждая самодвойственная функция на любой паре противоположных наборов принимает противоположные значения.

Докажем следующую лемму.

Лемма (о несамодвойственной функции)**.** *Пусть функция $f(x_1,\dots,x_n)$ не принадлежит множеству S. Тогда, подставляя в нее вместо переменных $x_1,\dots,x_n$ переменную x или ее отрицание $\overline{x}$, можно получить константу.*

Доказательство. Пусть $f(x_1,\dots,x_n)\notin S$. Тогда существует пара противоположных наборов $(\alpha_1,\dots,\alpha_n)$ и $(\overline{\alpha}_1,\dots,\overline{\alpha}_n)$, таких, что значения функции f на этих наборах равны, т. е.

$$f(\alpha_1,\dots,\alpha_n)=f(\overline{\alpha}_1,\dots,\overline{\alpha}_n).$$

Подставим в функцию f вместо переменных $x_1,\dots,x_n$ функции $x\oplus\alpha_1,\dots,x\oplus\alpha_n$ соответственно (т. е. вместо каждой переменной x_i подставляется либо x, либо $\overline{x}$). Получим некоторую функцию $h(x)=f(x\oplus\alpha_1,\dots,x\oplus\alpha_n)$. Так как $h(0)=f(\alpha_1,\dots,\alpha_n)$, а $h(1)=f(\overline{\alpha}_1,\dots,\overline{\alpha}_n)$, то выполняется равенство $h(0)=h(1)$. Поэтому функция h является константой.

Следствие. *Если $f\notin S$, то $0,1\in[\{f,\bar{}\}]$.*

Доказательство. Одну из констант это множество содержит по предыдущей лемме; вторая константа получается при помощи отрицания.

Лекция № 3

Определим правило сравнения наборов из нулей и единиц одинаковой длины. Будем говорить, что выполняется неравенство $(\alpha_1,\dots,\alpha_n) \leqslant (\beta_1,\dots,\beta_n)$, если $\alpha_1 \leqslant \beta_1,\dots,\alpha_n \leqslant \beta_n$. При этом мы считаем, что $0 \leqslant 0$, $0 \leqslant 1$, $1 \leqslant 1$ и $1 \not\leqslant 0$. Теперь мы можем упорядочить некоторые последовательности наборов из нулей и единиц: $(0,\dots,0) \leqslant \dots \leqslant (\alpha_1,\dots,\alpha_n) \leqslant \dots \leqslant (1,\dots,1)$. Отметим, что не все наборы одинаковой длины сравнимы, например $(0,1) \not\leqslant (1,0)$ и $(1,0) \not\leqslant (0,1)$.

Функция $f(x_1,\dots,x_n)$ называется *монотонной*, если для любых двух наборов из нулей и единиц $(\alpha_1,\dots,\alpha_n)$ и $(\beta_1,\dots,\beta_n)$, таких, что $(\alpha_1,\dots,\alpha_n) \leqslant (\beta_1,\dots,\beta_n)$, выполняется неравенство

$$f(\alpha_1,\dots,\alpha_n) \leqslant f(\beta_1,\dots,\beta_n).$$

Множество всех монотонных функций алгебры логики обозначается через M.

Примеры.

1. Функции 1, 0, x, x_1x_2, $x_1 \vee x_2$ являются монотонными (это видно из табл. 3 лекции 1).
2. Функции $\overline{x}$, x_1/x_2 не являются монотонными.

Как видно из примеров, множество M нетривиально. Докажем, что оно замкнуто. Так как тождественная функция принадлежит M, то достаточно доказать, что если $f_0(x_1,\dots,x_n) \in M$ и $f_1,\dots,f_n \in M$, то и функция $f = f_0(f_1,\dots,f_n)$ принадлежит M. Будем считать, что $f_1,\dots,f_n$ — функции от одного и того же набора переменных $y_1,\dots,y_m$ (в противном случае недостающие переменные можно добавить в качестве несущественных). Пусть $(\alpha_1,\dots,\alpha_m)$ и $(\beta_1,\dots,\beta_m)$ — произвольные наборы, такие, что $(\alpha_1,\dots,\alpha_m) \leqslant (\beta_1,\dots,\beta_m)$. Так как функции $f_1(y_1,\dots,y_m),\dots,f_n(y_1,\dots,y_m)$ принадлежат M, то $f_i(\alpha_1,\dots,\alpha_m) \leqslant f_i(\beta_1,\dots,\beta_m)$ для всех $i = 1,\dots,n$. Поэтому

$$(f_1(\alpha_1,\dots,\alpha_m),\dots,f_n(\alpha_1,\dots,\alpha_m)) \leqslant$$

$$\leqslant (f_1(\beta_1,\dots,\beta_m),\dots,f_n(\beta_1,\dots,\beta_m)).$$

Так как f_0 монотонна, то

$$f(\alpha_1,\dots,\alpha_m) = f_0(f_1(\alpha_1,\dots,\alpha_m),\dots,f_n(\alpha_1,\dots,\alpha_m)) \leqslant$$

$$\leqslant f_0(f_1(\beta_1,\dots,\beta_m),\dots,f_n(\beta_1,\dots,\beta_m)) = f(\beta_1,\dots,\beta_m).$$

То есть $f \in M$.

Докажем следующую лемму.

Лемма 1 (о немонотонной функции). *Из любой немонотонной функции, подставляя вместо некоторых ее переменных константы, можно получить отрицание.*

Доказательство. Пусть $f(x_1,\dots,x_n) \notin M$. Тогда существуют два набора $(\alpha_1,\dots,\alpha_n)$ и $(\beta_1,\dots,\beta_n)$, такие, что $(\alpha_1,\dots,\alpha_n) \leqslant (\beta_1,\dots,\beta_n)$, а $f(\alpha_1,\dots,\alpha_n) \not\leqslant f(\beta_1,\dots,\beta_n)$, т. е. $f(\alpha_1,\dots,\alpha_n) = 1$, а $f(\beta_1,\dots,\beta_n) = 0$. Поэтому наборы $(\alpha_1,\dots,\alpha_n)$ и $(\beta_1,\dots,\beta_n)$ разные. Пусть они различаются в m разрядах, $m \geqslant 1$. Переставим компоненты этих наборов таким образом, чтобы на первые m мест попали те компоненты, в которых эти наборы различаются. В результате мы получим наборы $(0,\dots,0,\alpha_{m+1},\dots,\alpha_n)$ и $(1,\dots,1,\alpha_{m+1},\dots,\alpha_n)$ (так как $\alpha_{m+1} = \beta_{m+1},\dots,\alpha_n = \beta_n$), причем выполняются равенства $f(0,\dots,0,\alpha_{m+1},\dots,\alpha_n) = 1$, $f(1,\dots,1,\alpha_{m+1},\dots,\alpha_n) = 0$. Рассмотрим наборы $\tilde{\alpha}^{(0)},\dots,\tilde{\alpha}^{(m)}$, такие, что

$$\tilde{\alpha}^{(i)} = (\underbrace{1,\dots,1}_{i},\underbrace{0,\dots,0}_{m-i},\alpha_{m+1},\dots,\alpha_n).$$

Рассмотрим значения $f(\tilde{\alpha}^{(1)}),\dots,f(\tilde{\alpha}^{(m)})$ функции f на этих наборах.

Так как $f(\tilde{\alpha}^{(0)}) = f(0,\dots,0,\alpha_{m+1},\dots,\alpha_n) = 1$, а $f(\tilde{\alpha}^{(m)}) = f(1,\dots,1,\alpha_{m+1},\dots,\alpha_n) = 0$, то найдется такой номер j $(1 \leqslant j \leqslant m)$, что $f(\tilde{\alpha}^{(j-1)}) = 1$, $f(\tilde{\alpha}^{(j)}) = 0$. То есть

$$f(\underbrace{1,\dots,1}_{j-1},0,\underbrace{0,\dots,0}_{m-j},\alpha_{m+1},\dots,\alpha_n) = 1,$$

$$f(\underbrace{1,\dots,1}_{j-1},1,\underbrace{0,\dots,0}_{m-j},\alpha_{m+1},\dots,\alpha_n) = 0.$$

В наборах $\tilde{\alpha}^{(j-1)}$ и $\tilde{\alpha}^{(j)}$ все компоненты совпадают, кроме компоненты с номером j. Рассмотрим функцию $h(x) = f(1,\dots,1,x,0,\dots,0,\alpha_{m+1},\dots,\alpha_n)$. Очевидно, что $h(x) = \overline{x}$, что и требовалось получить. Итак, из функции f мы получили функцию $h(x) = \overline{x}$.

Следствие. *Если $f \notin M$, то $\overline{x} \in [\{f, 0, 1\}]$.*

Теперь мы в состоянии сформулировать и доказать одну из основных теорем алгебры логики.

Теорема (о функциональной полноте). *Система F функций алгебры логики полна тогда и только тогда, когда $F \not\subseteq T_0$, $F \not\subseteq T_1$, $F \not\subseteq L$, $F \not\subseteq S$, $F \not\subseteq M$.*

Доказательство. *Необходимость.* Если F содержится в каком-либо из этих множеств, например в M, то $[F] \subseteq [M] \neq P_2$, т. е. система F неполная.

Достаточность. Так как $F \not\subseteq T_0$, то найдется функция $f_0 \in F$, такая, что $f_0 \notin T_0$. Аналогично найдутся функции f_1, f_L, f_S, f_M из F, такие, что $f_1 \notin T_1$, $f_L \notin L$, $f_S \notin S$, $f_M \notin M$ (некоторые из них могут совпадать).

Разобьем дальнейшее доказательство на три этапа.

1. Получение констант. Возьмем функцию $f_0 \notin T_0$. Рассмотрим функцию $\varphi(x) = f_0(x, \ldots, x)$. Тогда $\varphi(0) = f_0(0, \ldots, 0) = 1$. Если $\varphi(1) = 1$, то функция φ — константа 1, а функция $\psi(x) = f_1(\varphi(x), \ldots, \varphi(x)) = f_1(1, \ldots, 1) = 0$ — константа 0. Если $\varphi(1) = 0$, то $\varphi(x) = \overline{x}$. Тогда по лемме о несамодвойственной функции при помощи отрицания из f_S можно получить обе константы, т. е. $0, 1 \in [\{f_S, \bar{}\}] \subseteq [F]$. Таким образом, мы получим обе константы.
2. Получение отрицания. По лемме о немонотонной функции при помощи констант из функции f_M можно получить отрицание, т. е. $\overline{x} \in [\{f_M, 0, 1\}] \subseteq [F]$.
3. Получение конъюнкции. По лемме о нелинейной функции при помощи констант, отрицания и функции f_L можно получить конъюнкцию, т. е. $x_1 x_2 \in [\{f_L, 0, 1, \bar{}\}] \subseteq [F]$.

Итак, мы получили, что множество $[F]$ содержит полную подсистему. Следовательно, система F полная.

Следствие. *Из любой полной системы можно выделить полную подсистему, состоящую не более чем из пяти функций.*

Доказательство. Для доказательства достаточно взять из системы F функции f_0, f_1, f_L, f_S и f_M.

Покажем, что на самом деле можно всегда обойтись не более чем четырьмя функциями. Вернемся к первому шагу доказательства теоремы (получение констант). В первом случае, когда $\varphi(x) = 1$, функция $\varphi(x)$ является несамодвойственной и, следовательно, можно обойтись без функции f_S. Во втором случае, когда $\varphi(x) = \overline{x}$, функция $\varphi(x)$ является немонотонной и поэтому можно обойтись без функции f_M. Таким образом, из любой полной системы можно выделить полную подсистему, состоящую не более чем из четырех функций.

Покажем, что в общем случае это число уменьшить нельзя.

Рассмотрим систему $F = [0, 1, x_1x_2, x_1 \oplus x_2 \oplus x_3]$. Составим таблицу, в которой укажем, каким классам из T_0, T_1, L, S, M принадлежат функции системы F. Будем ставить знак $''+''$, если функция принадлежит соответствующему классу, и знак $''-''$, если не принадлежит (см. табл. 1).

Таблица 1

	T_0	T_1	L	S	M
0	+	−	+	−	+
1	−	+	+	−	+
x_1x_2	+	+	−	−	+
$x_1 \oplus x_2 \oplus x_3$	+	+	+	+	−

В этой таблице все очевидно, кроме, может быть, того, что функция $x_1 \oplus x_2 \oplus x_3$ является самодвойственной и не является монотонной. Эта функция самодвойственная, так как выполняется равенство

$$(x_1 \oplus x_2 \oplus x_3)^* = x_1 \oplus 1 \oplus x_2 \oplus 1 \oplus x_3 \oplus 1 \oplus 1 = x_1 \oplus x_2 \oplus x_3,$$

и немонотонная, поскольку из нее подстановкой констант можно получить отрицание: $\overline{x} = 1 \oplus 0 \oplus x$.

По предыдущей теореме система F полная, так как для любого из пяти классов в F найдется функция, которая не принадлежит этому классу. Однако ни одну функцию из этой системы удалить нельзя, так как все функции, кроме константы 0, принадлежат классу T_1, все функции, кроме константы 1, принадлежат классу T_0, все функции, кроме x_1x_2, принадлежат классу L, все функции, кроме $x_1 \oplus x_2 \oplus x_3$, принадлежат классу M.

Рассмотрим семейство $\mathcal{F} = \{T_0, T_1, L, S, M\}$. Покажем, что ни один из классов этого семейства не содержится в другом.

Лемма 2. *Для любых двух различных классов из семейства $\mathcal{F}$ существует функция, принадлежащая одному из них и не принадлежащая другому.*

Доказательство. Составим таблицу из пяти строк и пяти столбцов. Каждой строке и каждому столбцу соответствуют классы T_0, T_1, L, S, M. В клетку таблицы, которая стоит на пересечении i-й строки и j-го столбца, $i \neq j$, поместим функцию, принадлежащую классу, который соответствует i-й строке, и не принадлежащую классу, который соответствует j-му столбцу. Клетки таблицы, стоящие на главной диагонали, оставим пустыми (см. табл. 2).

Таблица 2

	T_0	T_1	L	S	M
T_0		0	x_1x_2	0	$x_1 \oplus x_2$
T_1	1		x_1x_2	1	$x_1 \oplus x_2 \oplus x_3$
L	1	0		1	$x_1 \oplus x_2$
S	$\overline{x}$	$\overline{x}$	$\psi(x_1, x_2, x_3)$		$\overline{x}$
M	1	0	x_1x_2	1	

В этой таблице функция $\psi(x_1, x_2, x_3)$ определяется равенством

$$\psi(x_1, x_2, x_3) = x_1(x_2 \vee x_3) \vee x_2x_3.$$

Легко видеть, что ψ — самодвойственная функция (см. табл. 3); функция ψ нелинейная, так как подстановкой константы 0 мы из нее можем получить конъюнкцию: $\psi(x_1, x_2, 0) = x_1x_2$.

Таблица 3

x_1	x_2	x_3	$\psi(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Теорема. *Пусть A — замкнутый класс, такой, что $A \neq P_2$. Тогда A содержится в одном из пяти классов T_0, T_1, L, S, M.*

Доказательство. Допустим, что A не содержится ни в одном из этих пяти классов. Тогда по предыдущей теореме A — полная система, т. е. $A = [A] = P_2$, что противоречит условию.

Множество булевых функций F называется *предполным классом*, если выполняются следующие условия:

1) $F \neq P_2$;
2) $F = [F]$;
3) для любой функции f, такой, что $f \notin F$, система $F \cup \{f\}$ является полной.

Теорема. *В P_2 существует только пять предполных классов: T_0, T_1, L, S, M.*

Доказательство. Докажем сначала, что любой из этих пяти классов является предполным. Рассмотрим, например, класс S. Очевидно, что первые два условия выполнены. По лемме 2 множество S содержит функции f_0, f_1, f_L, f_M, которые не принадлежат классам T_0, T_1, L, M соответственно. Поэтому для любой функции f_S, такой, что $f_S \notin S$, в системе $S \cup \{f_S\}$ найдутся функции, не принадлежащие ни одному из этих пяти классов. Следовательно, $S \cup \{f_S\}$ — полная система. Поэтому S — предполный класс.

Докажем теперь, что других предполных классов нет. Пусть A — замкнутый класс, отличный от P_2 и не совпадающий ни с одним из пяти рассматриваемых классов. Тогда по предыдущей теореме A содержится в каком-то из этих классов. Пусть, например, $A \subseteq L$. Так как по предположению $A \neq L$, то имеем строгое включение $A \subset L$. Тогда найдется функция f, такая, что $f \in L$ и $f \notin A$. То есть $A \cup \{f\} \subseteq L$. Поэтому $[A \cup \{f\}] \subseteq L \neq P_2$. Следовательно, система $A \cup \{f\}$ неполная. Поэтому (так как не выполняется свойство 3) класс A не является предполным.

Замкнутые классы булевых функций были описаны американским математиком Э. Постом. Он изучил ряд важных свойств этих классов. Расскажем коротко об основных результатах Поста.

Пусть F — произвольный замкнутый класс булевых функций, и пусть A — некоторая система функций, содержащихся в F. Система A называется *полной* в F, если $[A] = F$. Система A называется *базисом* класса F, если она полна в F, но всякая ее собственная подсистема не является полной в F.

Теорема. *Множество замкнутых классов алгебры логики имеет счетную мощность.*

Теорема. *Каждый замкнутый класс булевых функций имеет конечный базис.*

В нашем курсе эти теоремы Поста доказываться не будут. Более подробные сведения о функциях алгебры логики содержатся, например, в книге [1)].

Упомянем еще об одном важном свойстве функций алгебры логики.

Лемма 3. *Пусть функции* $f(x_1, x_2, \ldots, x_n)$ *и* $g(y_1, y_2, \ldots, y_m)$ *существенно зависят от всех своих переменных. Тогда функция*

$$h(x_1, x_2, \ldots, x_{n-1}, y_1, y_2, \ldots, y_m) =$$

$$= f(x_1, x_2, \ldots, x_{n-1}, g(y_1, y_2, \ldots, y_m))$$

также существенно зависит от всех переменных.

Доказательство. Легко видеть, что все переменные $x_1, x_2, \ldots, x_{n-1}$ и все переменные $y_1, y_2, \ldots, y_m$ равноправны между собой. Поэтому нам достаточно показать, что h существенно зависит от переменных x_1 и y_m. Так как по условию f существенно зависит от x_1, то существуют наборы $(0, \alpha_2, \ldots, \alpha_n)$ и $(1, \alpha_2, \ldots, \alpha_n)$, такие, что $f(0, \alpha_2, \ldots, \alpha_n) \neq f(1, \alpha_2, \ldots, \alpha_n)$. Так как g существенно зависит от всех переменных, то она не константа. Поэтому найдется набор $(\beta_1, \ldots, \beta_m)$, на котором выполняется равенство $g(\beta_1, \ldots, \beta_m) = \alpha_n$. Но тогда

$$h(0, \alpha_2, \ldots, \alpha_n, \beta_1, \ldots, \beta_m) \neq h(1, \alpha_2, \ldots, \alpha_n, \beta_1, \ldots, \beta_m).$$

То есть h существенно зависит от переменной x_1. Далее, так как f существенно зависит от x_n, то существуют наборы $(\alpha_1, \ldots, \alpha_{n-1}, 0)$ и $(\alpha_1, \ldots, \alpha_{n-1}, 1)$, такие, что

$$f(\alpha_1, \ldots, \alpha_{n-1}, 0) \neq f(\alpha_1, \ldots, \alpha_{n-1}, 1).$$

Так как g существенно зависит от y_m, то найдутся наборы $(\beta_1, \ldots, \beta_{m-1}, 0)$ и $(\beta_1, \ldots, \beta_{m-1}, 1)$, такие, что

$$g(\beta_1, \ldots, \beta_{m-1}, 0) \neq g(\beta_1, \ldots, \beta_{m-1}, 1).$$

Поэтому

$$h(\alpha_1, \ldots, \alpha_{n-1}, \beta_1, \ldots, \beta_{m-1}, 0) \neq h(\alpha_1, \ldots, \alpha_{n-1}, \beta_1, \ldots, \beta_{m-1}, 1).$$

А значит, y_m— существенная переменная функции h.

[1)] См.: *Яблонский С.В., Гаврилов Г.П., Кудрявцев В.Б.* Функции алгебры логики и классы Поста. М.: Наука, 1966.

ФУНКЦИИ k-ЗНАЧНОЙ ЛОГИКИ

Лекция № 4

На предыдущих лекциях мы определили функции алгебры логики и изучили ряд свойств этих функций. Аналогичным образом можно определить функции k-значной логики. Значения переменных и самих функций берутся из множества $E_k = \{0, 1, \ldots, k-1\}$, $k \geqslant 3$. Множество всех таких функций обозначается через P_k. Как и булевы функции, каждую функцию $f(x_1, \ldots, x_n)$ из P_k можно задать таблицей (см. табл. 1).

Таблица 1

x_1	$\ldots$	x_n	$f(x_1, \ldots, x_n)$
0	$\ldots$	0	$f(0, \ldots, 0)$
	$\ldots$		$\ldots$
σ_1	$\ldots$	σ_n	$f(\sigma_1, \ldots, \sigma_n)$
	$\ldots$		$\ldots$
$k-1$	$\ldots$	$k-1$	$f(k-1, \ldots, k-1)$

Пусть $p_k(n)$ — число всех функций $f(x_1, \ldots, x_n)$ из P_k. Количество различных наборов значений переменных равно k^n. На каждом из этих наборов функция $f(x_1, \ldots, x_n)$ может принимать любое из k значений. Следовательно, всего таких функций будет $p_k(n) = k^{k^n}$. Это число очень быстро растет, например уже в P_3 число функций от переменных x_1 и x_2 равно $p_3(2) = 19\,683$. Все основные понятия, такие, как формула над множеством функций, значение формулы на наборе значений переменных, функция, реализуемая формулой, существенная и несущественная переменные и др., вводятся точно так же, как и в двузначной логике (определения почти дословно повторяются), и мы не будем их воспроизводить. Однако не следует забывать, что переменные и функции принимают уже не два значения, а больше. В частности, если известно значение x из E_k, то нельзя определить значение y из E_k только на основе соотношения $y \neq x$ ($k \geqslant 3$). Это приводит к принципиальным отличиям P_k, $k \geqslant 3$, от P_2.

В конце предыдущей лекции мы показали, что при подстановке одной булевой функции в другую сохраняется существенная за-

висимость от переменных. Покажем, что для функций k-значной логики при $k \geqslant 3$ аналогичное утверждение неверно.

Рассмотрим функцию $\varphi(x_1, x_2)$, заданную табл. 2.

Таблица 2

x_1	0	1	2
0	0	0	0
1	0	0	0
2	0	0	1

Функция φ принадлежит P_3 и принимает ненулевое значение только на наборе $(2, 2)$. Поэтому функция $\varphi(x, \varphi(y, z))$ — константа 0, поскольку для любых $\beta, \gamma \in E_3$ выполняется неравенство $\varphi(\beta, \gamma) \neq 2$.

Рассмотрим следующие "элементарные" функции k-значной логики.

1. Константы $0, 1, \ldots, k-1$.
2. Тождественная функция x.
3. Функции $\overline{x} = x + 1 \pmod k$, $N(x) = k - 1 - x$. Эти функции являются обобщениями отрицания в P_2. Функция $N(x)$ является "зеркальным" отражением x. Она обозначается также $\sim x$ и называется *отрицанием Лукашевича*.
4. Функции $I_i(x)$ и $j_i(x)$, $i = 0, 1, \ldots, k-1$:

$$I_i(x) = \begin{cases} k-1, \text{ если } x = i; \\ 0, \text{ если } x \neq i, \end{cases}$$

$$j_i(x) = \begin{cases} 1, \text{ если } x = i; \\ 0, \text{ если } x \neq i. \end{cases}$$

   Эти функции являются аналогами функции x^σ в P_2.
5. Функции $\min(x_1, x_2)$ и $x_1 x_2 \pmod k$. Эти функции являются обобщением конъюнкции. Функция $\min(x_1, x_2)$ обозначается также $x_1 \& x_2$.
6. Функция $\max(x_1, x_2)$. Она является аналогом дизъюнкции в P_2 и обозначается также $x_1 \vee x_2$.
7. Функция $x_1 + x_2 \pmod k$.

Эти функции, так же как и их аналоги в P_2, обладают рядом важных свойств. Например, функции $\max(x_1, x_2)$, $x_1 x_2 \pmod k$, $\min(x_1, x_2)$ и $x_1 + x_2 \pmod k$ обладают свойствами коммутативности и ассоциативности, имеет место дистрибутивность $\max(x_1, x_2)$ относительно $\min(x_1, x_2)$ и так далее. Эти свойства позволяют вводить некоторые соглашения, упрощающие вид формул, например:

$$\&_{i=1}^{n} A_i = \min(A_1, A_2, \ldots, A_n),$$

$$\bigvee_{i=1}^{n} A_i = \max(A_1, A_2, \ldots, A_n).$$

Система F функций k-значной логики называется *полной*, если любая функция из P_k выражается формулой над F.

Теорема 1. *Система*

$$F = \{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$$

является полной.

Доказательство. Для функций k-значной логики имеет место представление, которое является аналогом совершенной дизъюнктивной нормальной формы:

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_n)} I_{\sigma_1}(x_1) \& \ldots \& I_{\sigma_n}(x_n) \& f(\sigma_1, \ldots, \sigma_n),$$

где максимум берется по всевозможным наборам значений переменных $(x_1, \ldots, x_n)$. Действительно, рассмотрим произвольный набор $(\alpha_1, \ldots, \alpha_n)$. Найдем значение формулы на этом наборе. Если выполняется равенство $(\sigma_1, \ldots, \sigma_n) = (\alpha_1, \ldots, \alpha_n)$, то

$$I_{\sigma_1}(\alpha_1) \& \ldots \& I_{\sigma_n}(\alpha_n) \& f(\sigma_1, \ldots, \sigma_n) = f(\alpha_1, \ldots, \alpha_n),$$

так как $I_{\sigma_1}(\alpha_1) = \cdots = I_{\sigma_n}(\alpha_n) = k - 1$ (т. е. равны максимальному значению из множества E_k). Если же $(\sigma_1, \ldots, \sigma_n) \neq (\alpha_1, \ldots, \alpha_n)$, то найдется i $(1 \leqslant i \leqslant n)$, такое, что $\sigma_i \neq \alpha_i$. Тогда $I_{\sigma_i}(\alpha_i) = 0$ (равно наименьшему значению из E_k). Поэтому

$$I_{\sigma_1}(\alpha_1) \& \ldots \& I_{\sigma_n}(\alpha_n) \& f(\sigma_1, \ldots, \sigma_n) = 0.$$

Следовательно,

$$\bigvee_{(\sigma_1, \ldots, \sigma_n)} I_{\sigma_1}(\alpha_1) \& \ldots \& I_{\sigma_n}(\alpha_n) \& f(\sigma_1, \ldots, \sigma_n) = f(\alpha_1, \ldots, \alpha_n).$$

Поскольку рассматриваемая формула построена из функций системы F, то F — полная система.

Следствие. *Для любого $k \geqslant 3$ в P_k существуют конечные полные системы.*

Для функций k-значной логики, как и для булевых функций, имеет место следующее утверждение (достаточное условие полноты).

Утверждение 1. *Если F — полная система и любая функция из F реализуется формулой над G, то G — полная система.*

Доказательство аналогично доказательству соответствующего утверждения для P_2.

Из этого утверждения и теоремы получаем

Следствие. *Из любой полной системы функций k-значной логики можно выделить конечную полную подсистему.*

Используя достаточное условие полноты, приведем примеры других полных систем в P_k.

Утверждение 2. *Система* $\{\max(x_1, x_2), x+1 \pmod k\}$ *является полной.*

Доказательство. Разобьем доказательство утверждения на несколько этапов.

Построим сначала константы. Рассмотрим функции

$$\overline{x} = x+1,\ (\overline{x+1}) = x+2, \ldots, (\overline{x+k-2}) = x+k-1,\ x+k = x.$$

При каждом значении x из $E_k = \{0, 1, \ldots, k-1\}$ множество значений, принимаемых всеми этими функциями, равно E_k. Поэтому

$$\max(x+1, x+2, \ldots, x+k-1, x) = k-1.$$

Остальные константы получаются при помощи функции $x+1 \pmod k$.

Построим затем функции $I_i(x)$, $i = 0, 1, \ldots, k-1$. Рассмотрим функцию

$$\varphi(x) = \max(x, x+1, \ldots, x+k-2) + 1.$$

Если $x = 0$, то $\varphi(0) = \max(0, 1, \ldots, k-2) + 1 = k-1$. Если же $x = \sigma \neq 0$, то среди чисел σ, $\sigma+1, \ldots, \sigma+k-2$ есть число $k-1$. Поэтому $\varphi(\sigma) = k-1+1 = 0$. То есть

$$\varphi(x) = \begin{cases} k-1, \text{ если } x = 0; \\ 0, \text{ если } x \neq 0, \end{cases}$$

а значит, $\varphi(x) = I_0(x)$. Аналогично функция

$$\psi(x) = \max_{\alpha \neq k-1-i} \{x+\alpha\} + 1$$

равна функции $I_i(x)$. Действительно, при $x = i$ выполняется равенство

$$\psi(i) = \max_{\alpha \neq k-1-i} \{i+\alpha\} + 1 = \max_{\alpha+i \neq k-1} \{i+\alpha\} + 1 = k-2+1 = k-1.$$

А при $x = \sigma \neq i$ среди чисел

$$\sigma,\ \sigma+1, \ldots, \sigma+k-1-(i-1),\ \sigma+k-1-(i+1), \ldots, \sigma+k-1$$

есть число $k-1$. Поэтому $\psi(\sigma) = k-1+1 = 0$. То есть $\psi(x) = I_i(x)$.

Для доказательства утверждения нам осталось получить только функцию $\min(x_1, x_2)$. Для этого воспользуемся следующим равенством:

$$\min(x_1, x_2) = N(\max(N(x_1), N(x_2))),$$

которое аналогично равенству $x_1 \& x_2 = \overline{\overline{x}_1 \vee \overline{x}_2}$ для функций алгебры логики. Таким образом, нам достаточно получить функцию $N(x)$.

Покажем, как получать произвольные функции одной переменной из P_k. Для произвольных α, β из E_k рассмотрим функции

$$\varphi_{\alpha,\beta}(x) = \begin{cases} \beta, \text{ если } x = \alpha; \\ 0, \text{ если } x \neq \alpha, \end{cases}$$

$$\psi(x) = \max(I_\alpha(x), k-1-\beta).$$

При $x = \alpha$ функция ψ принимает значение $k-1$, а при $x = \gamma \neq \alpha$ выполняется равенство $\psi(\gamma) = k-1-\beta$. Поэтому $\varphi_{\alpha,\beta}(x) = \psi(x) + \beta + 1$. Пусть теперь $g(x)$ — произвольная функция (одной переменной) из P_k. Тогда

$$g(x) = \max(\varphi_{0,g(0)}(x), \varphi_{1,g(1)}(x), \ldots, \varphi_{k-1,g(k-1)}(x)).$$

Следовательно, мы можем получить и функцию $N(x)$.

Итак, мы выразили каждую функцию системы

$$F = \{0, 1, \ldots, k-1, I_0(x), \ldots, I_{k-1}(x), \min(x_1, x_2), \max(x_1, x_2)\}$$

в виде формул над исходной системой. Поскольку система F является полной, то в силу достаточного условия полноты получаем, что $\{\max(x_1, x_2), x + 1 \pmod k\}$ — полная система.

Приведем пример полной системы в P_k, состоящей из одной функции. Положим

$$V_k(x_1, x_2) = \max(x_1, x_2) + 1.$$

Эта функция называется *функцией Вебба*. Она является аналогом штриха Шеффера.

Утверждение 3. *Система $\{V_k(x_1, x_2)\}$ является полной.*

Доказательство. Действительно, $V_k(x, x) = x + 1$. Поэтому можно получить любую функцию $\varphi(x) = x + c$, $c \in E_k$. Кроме того, выполняется равенство $V_k(x_1, x_2) + k - 1 = \max(x_1, x_2)$. Поскольку полученные функции образуют полную систему, то и система $\{V_k(x_1, x_2)\}$ является полной.

Пусть F — произвольное множество функций k-значной логики. *Замыканием* называется множество $[F]$ всех функций из P_k, выразимых формулами над F. Множество F называется *замкнутым*, если $F = [F]$. Замкнутые множества называются также замкнутыми классами. Таким образом, система F полная, если $[F] = P_k$. Множество F называется *предполным классом*, если выполнены следующие условия:

1) $F \neq P_k$;
2) $F = [F]$;
3) для любой функции f, такой, что $f \notin F$, система $F \cup \{f\}$ является полной.

Имеет место следующая

Теорема 2 (А. В. Кузнецов). *Для любого k существует лишь конечное число предполных классов $M_1, M_2, \ldots, M_q$; при этом система F полна в P_k тогда и только тогда, когда $F \not\subseteq M_i$ для всех $i = 1, \ldots, q$.*

Доказательство этой теоремы будет приведено ниже.

Пусть $k \geqslant 2$. Обозначим через $\pi(k)$ число предполных классов в P_k. С. В. Яблонский показал, что при $k = 3$ их число равно 18. При росте k число $\pi(k)$ растет очень быстро (см. табл. 3).

Таблица 3

k	2	3	4	5	6
$\pi(k)$	3	18	80	547	15 267

Справедлива следующая асимптотическая формула:

$$\pi(k) \sim \delta(k) k 2^{C_{k-1}^{[\frac{k-1}{2}]}},$$

где C_n^m — число сочетаний из n элементов по m, а $\delta(k) = 1$, если k нечетно, и $\delta(k) = 2$, если k четно.

Полное описание предполных классов в P_k было получено И. Розенбергом в 1965 г. Более подробные сведения о предполных классах функций многозначной логики можно найти, например, в книге [1)].

В заключение приведем алгоритм распознавания полноты конечных систем функций в P_k.

Теорема 3. *Существует алгоритм распознавания полноты конечных систем функций в P_k.*

Доказательство. Пусть F — некоторое конечное множество функций k-значной логики, $F = \{f_1(x_1, \ldots, x_{n_1}, \ldots, f_t(x_1, \ldots, x_{n_t})\}$. Возьмем две переменные x_1 и x_2. Построим по индукции последовательность множеств $R_0, R_1, \ldots$, каждое из которых состоит из функций от переменных x_1 и x_2.

Положим $R_0 = \emptyset$. Пусть уже построены множества функций $R_0, R_1, \ldots, R_r$, и пусть $|R_r| = s_r$ ($s_r = 0$ при $r = 0$). Для каждого $j = 1, \ldots, t$ рассмотрим всевозможные формулы вида $f_j(A_1, \ldots, A_{n_j})$, где $A_1, \ldots, A_{n_j}$ — либо функции из множества R_r, либо переменные x_1 или x_2. Очевидно, что каждая такая формула реализует некоторую функцию от переменных x_1 и x_2. Обозначим множество таких функций через Δ_r. Легко видеть, что $|\Delta_r| \leqslant t(s_r + 2)^{n_j}$. Положим $R_{r+1} = R_r \cup \Delta_r$. Тогда

$$R_0 \subseteq R_1 \subseteq \ldots \subseteq R_r \subseteq R_{r+1} \subseteq \ldots .$$

Из построения следует, что если $R_{r+1} = R_r$, то все последующие классы совпадают с R_r. Таким образом, начиная с некоторого r^*

[1)] См.: *Яблонский С.В., Гаврилов Г.П., Набебин А.А.* Предполные классы в многозначных логиках. М.: Изд-во МЭИ, 1997.

процесс стабилизируется, поскольку каждый класс R_i содержит только функции от переменных x_1 и x_2, а всего таких функций $p_k(2) = k^{k^2}$. То есть $r^* \leqslant k^{k^2}$. Заметим,что множество R_{r^*} содержит все функции от переменных x_1 и x_2 из класса $[F]$. Поэтому система F полна тогда и только тогда, когда функция $V_k(x_1, x_2)$ принадлежит множеству R_{r^*}.

Следует отметить, что поскольку число $\pi(k)$ растет очень быстро с ростом k, то приведенный алгоритм является более наглядным (и менее трудоемким) по сравнению с аналогичным алгоритмом, построенным на основе приведенного выше критерия распознавания полноты в терминах предполных классов.

Лекция № 5

Приведем критерии полноты систем функций в P_k, основанные на некоторой дополнительной информации об исходных системах.

Функция $f(x_1, \ldots, x_n)$ из P_k называется *существенной*, если она существенно зависит не менее чем от двух переменных и принимает k значений.

Теорема 1 (критерий Е. Слупецкого). *Пусть система F из P_k, $k \geqslant 3$, содержит все функции одной переменной. Тогда F полна тогда и только тогда, когда F содержит существенную функцию.*

Теорема 2 (критерий С. В. Яблонского). *Пусть система F из P_k, $k \geqslant 3$, содержит все функции одной переменной, принимающие не более $k-1$ значений. Тогда F полна тогда и только тогда, когда F содержит существенную функцию.*

Критерий Яблонского является усилением критерия Слупецкого. Для доказательства этих утверждений мы установим необходимость условий теоремы 1 и достаточность условий теоремы 2.

Доказательство. *Необходимость* (*теорема* 1). Предположим, что F содержит все функции из P_k от одной переменной и не содержит существенных функций. Покажем, что F — неполная система. Действительно, каждая формула Φ над F, которая реализует некоторую функцию, существенно зависящую более чем от одной переменной, имеет вид

$$\Phi = f_{i_1}(f_{i_2}(\ldots f_{i_m}(g(A_1, \ldots, A_n))\ldots)),$$

где $f_{i_1}, \ldots, f_{i_m}$ — функции одной переменной, g — функция из F, существенно зависящая от $n \geqslant 2$ переменных, а $A_1, \ldots, A_n$ — либо переменные, либо формулы над F. Так как функция g принадлежит F, то она принимает не более $k-1$ значений. Поэтому формула Φ также принимает не более $k-1$ значений. Следовательно, формулы над F не могут реализовывать существенных функций. Таким образом, F — неполная система.

Нам понадобятся следующие леммы.

Лемма (о трех наборах). *Пусть $f(x_1, \ldots, x_n)$ — функция из P_k, существенно зависящая не менее чем от двух переменных и принимающая не менее трех значений. Пусть x_1 — ее существенная переменная. Тогда существуют три набора $(\alpha, \alpha_2, \ldots, \alpha_n)$,*

$(\beta, \alpha_2, \ldots, \alpha_n)$ и $(\alpha, \gamma_2, \ldots, \gamma_n)$, такие, что функция f на этих наборах принимает три попарно различных значения.

Доказательство. Поскольку функция $f(x_1, \ldots, x_n)$ существенно зависит от переменной x_1, то существуют наборы $(\alpha, \alpha_2, \ldots, \alpha_n)$ и $(\beta, \alpha_2, \ldots, \alpha_n)$, такие, что

$$f(\alpha, \alpha_2, \ldots, \alpha_n) \neq f(\beta, \alpha_2, \ldots, \alpha_n).$$

Рассмотрим множество M наборов следующего вида:

$$M = \{(0, \alpha_2, \ldots, \alpha_n), (1, \alpha_2, \ldots, \alpha_n), \ldots, (k-1, \alpha_2, \ldots, \alpha_n)\}$$

и множество M_f значений функции f на этих наборах:

$$M_f = \{f(0, \alpha_2, \ldots, \alpha_n), f(1, \alpha_2, \ldots, \alpha_n), \ldots, f(k-1, \alpha_2, \ldots, \alpha_n)\}.$$

Множество M_f содержит не менее двух элементов. Рассмотрим два случая.

Пусть $|M_f| = 2$, и пусть $M_f = \{\varepsilon_1, \varepsilon_2\}$. Так как функция f принимает по крайней мере три значения, то найдется набор $(\gamma, \gamma_2, \ldots, \gamma_n)$, такой, что $f(\gamma, \gamma_2, \ldots, \gamma_n) \notin M_f$. Рассмотрим набор $(\gamma, \alpha_2, \ldots, \alpha_n)$. Он принадлежит множеству M. Поэтому $f(\gamma, \alpha_2, \ldots, \alpha_n) \in \{\varepsilon_1, \varepsilon_2\}$. Пусть, например, выполняется равенство $f(\gamma, \alpha_2, \ldots, \alpha_n) = \varepsilon_1$. Тогда существует набор $(\delta, \alpha_2, \ldots, \alpha_n) \in M$, такой,что $f(\delta, \alpha_2, \ldots, \alpha_n) = \varepsilon_2$. Таким образом, наборы $(\delta, \alpha_2, \ldots, \alpha_n)$, $(\gamma, \alpha_2, \ldots, \alpha_n)$ и $(\gamma, \gamma_2, \ldots, \gamma_n)$ удовлетворяют условию леммы.

Пусть теперь $|M_f| \geqslant 3$. Так как f существенно зависит по крайней мере от двух переменных, то найдется такое число $\alpha \in E_k$, что функция $f(\alpha, x_2, \ldots, x_n)$ не является константой. Поэтому существует набор $(\gamma_2, \ldots, \gamma_n)$, такой, что

$$\varepsilon_1 = f(\alpha, \gamma_2, \ldots, \gamma_n) \neq f(\alpha, \alpha_2, \ldots, \alpha_n) = \varepsilon_2.$$

Так как $|M_f| \geqslant 3$, то найдется элемент $\beta \in E_k \setminus \{\alpha\}$, такой, что $f(\beta, \alpha_2, \ldots, \alpha_n) \in M_f \setminus \{\varepsilon_1, \varepsilon_2\}$. Таким образом, на наборах $(\alpha, \alpha_2, \ldots, \alpha_n)$, $(\beta, \alpha_2, \ldots, \alpha_n)$ и $(\alpha, \gamma_2, \ldots, \gamma_n)$ функция f принимает три разных значения. Лемма доказана.

Заметим, что эти наборы выбраны таким образом, что матрица A, состоящая из трех строк и n столбцов и составленная из этих наборов, в каждом столбце содержит не более двух различных элементов:

$$A = \begin{pmatrix} \alpha\, \alpha_2 \ldots\, \alpha_n \\ \beta\, \alpha_2 \ldots\, \alpha_n \\ \alpha\, \gamma_2 \ldots\, \gamma_n \end{pmatrix}.$$

Отметим также, что $(\alpha_2, \ldots, \alpha_n) \neq (\gamma_2, \ldots, \gamma_n)$.

Квадратом называется система из четырех наборов вида

$$\begin{aligned} \tilde{a}_1 &= (\alpha_1, \ldots, \alpha_{i-1}, \alpha, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta, \alpha_{j+1}, \ldots, \alpha_n), \\ \tilde{a}_2 &= (\alpha_1, \ldots, \alpha_{i-1}, \gamma, \alpha_{i+1}, \ldots, \alpha_{j-1}, \beta, \alpha_{j+1}, \ldots, \alpha_n), \\ \tilde{b}_1 &= (\alpha_1, \ldots, \alpha_{i-1}, \alpha, \alpha_{i+1}, \ldots, \alpha_{j-1}, \delta, \alpha_{j+1}, \ldots, \alpha_n), \\ \tilde{b}_2 &= (\alpha_1, \ldots, \alpha_{i-1}, \gamma, \alpha_{i+1}, \ldots, \alpha_{j-1}, \delta, \alpha_{j+1}, \ldots, \alpha_n), \end{aligned}$$

где $\alpha \neq \gamma$, $\beta \neq \delta$.

Квадрат $K = \{\tilde{a}_1, \tilde{a}_2, \tilde{b}_1, \tilde{b}_2\}$ удобно представлять в виде геометрического квадрата, в вершинах которого расположены наборы $\tilde{a}_1, \tilde{a}_2, \tilde{b}_1, \tilde{b}_2$; при этом в соседних вершинах находятся наборы, различающиеся ровно в одном разряде (см. рис. 1).

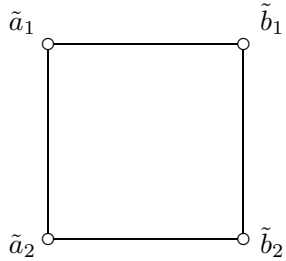


Рис. 1

Лемма (о квадрате). *Пусть $f(x_1, \ldots, x_n)$ — функция из P_k, существенно зависящая не менее чем от двух переменных и принимающая не менее трех значений. Тогда существует квадрат, на котором функция f некоторое значение принимает ровно один раз.*

Доказательство. Функция $f(x_1, \ldots, x_n)$ удовлетворяет условиям леммы о трех наборах. Поэтому существуют три набора $\tilde{a}_1 = (\alpha, \alpha_2, \ldots, \alpha_n)$, $\tilde{a}_2 = (\beta, \alpha_2, \ldots, \alpha_n)$ и $\tilde{c} = (\alpha, \gamma_2, \ldots, \gamma_n)$, на которых функция f принимает три попарно различных значения ε_1, ε_2 и ε_3 соответственно. Возьмем также набор $\tilde{d} = (\beta, \gamma_2, \ldots, \gamma_n)$. Без ограничения общности будем считать, что наборы $(\alpha_2, \ldots, \alpha_n)$

и $(\gamma_2,\dots,\gamma_n)$ различаются в первых $m-1$ разрядах, т. е. $\gamma_i \neq \alpha_i$ для всех $i = 2,\dots,m$; $\gamma_{m+1} = \alpha_{m+1},\dots,\gamma_n = \alpha_n$, $m \geqslant 2$. Таким образом, наборы $\tilde a_1$, $\tilde a_2$, $\tilde c$ и $\tilde d$ имеют вид

$$\tilde a_1 = (\alpha,\alpha_2,\alpha_3,\dots,\alpha_m,\tilde\alpha^{m+1}),$$

$$\tilde a_2 = (\beta,\alpha_2,\alpha_3,\dots,\alpha_m,\tilde\alpha^{m+1}),$$

$$\tilde c = (\alpha,\gamma_2,\gamma_3,\dots,\gamma_m,\tilde\alpha^{m+1}),$$

$$\tilde d = (\beta,\gamma_2,\gamma_3,\dots,\gamma_m,\tilde\alpha^{m+1}),$$

где $\tilde\alpha^{m+1} = (\alpha_{m+1},\dots,\alpha_n)$. Очевидно, что при $m = 2$ эти наборы образуют искомый квадрат, так как обязательно найдется значение из множества $\{\varepsilon_1,\varepsilon_2,\varepsilon_3\}$, которое f принимает ровно один раз.

Пусть $m \geqslant 3$. Построим последовательность квадратов $K_1, K_2,\dots,K_{m-1}$, такую, что $\tilde a_1,\tilde a_2 \in K_1$, $\tilde c \in K_{m-1}$ и при всех $i = 1,\dots,m-2$ множество $K_i \cap K_{i+1}$ состоит из двух наборов.

Для всех $i = 1,\dots,m-2$ определим наборы $\tilde b_1^i$, $\tilde b_2^i$ следующим образом. Положим

$$\tilde b_1^1 = (\alpha,\gamma_2,\tilde\alpha^2), \quad \tilde b_2^1 = (\beta,\gamma_2,\tilde\alpha^2),$$

а при $i \geqslant 2$

$$\tilde b_1^i = (\alpha,\tilde\gamma^i,\gamma_{i+1},\tilde\alpha^{i+2}), \quad \tilde b_2^i = (\beta,\tilde\gamma^i,\gamma_{i+1},\tilde\alpha^{i+2}),$$

где $\tilde\gamma^i = (\gamma_2,\dots,\gamma_i)$, $\tilde\alpha^{i+2} = (\alpha_{i+2},\dots,\alpha_n)$. Положим

$$K_1 = \{\tilde a_1,\tilde a_2,\tilde b_1^1,\tilde b_2^1\},\dots,$$

$$K_i = \{\tilde b_1^{i-1},\tilde b_2^{i-1},\tilde b_1^i,\tilde b_2^i\},\dots,$$

$$K_{m-1} = \{\tilde b_1^{m-2},\tilde b_2^{m-2},\tilde b_1^{m-1},\tilde b_2^{m-1}\}.$$

Каждое из этих множеств является квадратом. Действительно, множество K_1 состоит из наборов

$$\tilde a_1 = (\alpha,\alpha_2,\tilde\alpha^2), \quad \tilde b_1^1 = (\alpha,\gamma_2,\tilde\alpha^2),$$

$$\tilde a_2 = (\beta,\alpha_2,\tilde\alpha^2), \quad \tilde b_2^1 = (\beta,\gamma_2,\tilde\alpha^2),$$

а потому является квадратом. Аналогично множество K_i состоит из наборов

$$\tilde b_1^{i-1} = (\alpha,\tilde\gamma^i,\alpha_{i+1},\tilde\alpha^{i+2}), \quad \tilde b_1^i = (\alpha,\tilde\gamma^i,\gamma_{i+1},\tilde\alpha^{i+2}),$$

$$\tilde b_2^{i-1} = (\beta,\tilde\gamma^i,\alpha_{i+1},\tilde\alpha^{i+2}), \quad \tilde b_2^i = (\beta,\tilde\gamma^i,\gamma_{i+1},\tilde\alpha^{i+2})$$

и потому также является квадратом при всех $i = 2,\dots,m-1$. Кроме того, по построению выполняется равенство $K_i \cap K_{i+1} = \{\tilde b_1^i,\tilde b_2^i\}$ для всех $i = 1,\dots,m-2$, наборы $\tilde a_1$ и $\tilde a_2$ принадлежат квадрату K_1, а квадрат K_{m-1} содержит набор $\tilde b_1^{m-1} = (\alpha,\tilde\gamma^{m-1},\gamma_m,\tilde\alpha^{m+1}) = (\alpha,\gamma_2,\dots,\gamma_m,\alpha_{m+1},\dots,\alpha_n) = \tilde c$. Таким образом, искомая последовательность квадратов построена (см. рис. 2). Она представлена в виде последовательности геометрических квадратов, в вершинах которых расположены принадлежащие этим квадратам наборы.

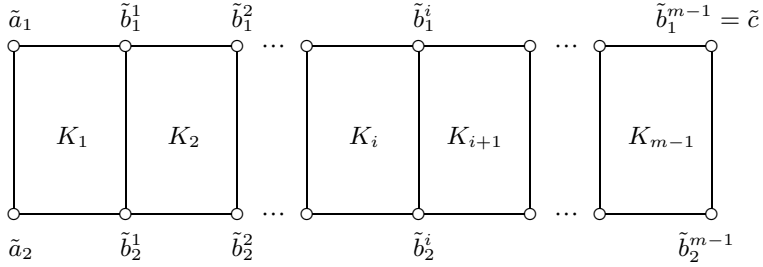


Рис. 2

Обозначим через A_i множество $\{f(\tilde b_1^i), f(\tilde b_2^i)\}$, $i \geqslant 1$, и положим $A_0 = \{f(\tilde a_1), f(\tilde a_2)\}$. Так как $f(\tilde b_1^{m-1}) = f(\tilde c) \notin \{\varepsilon_1,\varepsilon_2\} = A_0$, то $A_{m-1} \neq A_0$. Поэтому найдется такое число s $(0 \leqslant s \leqslant m-2)$, что $A_0 = A_1 = \dots = A_s \neq A_{s+1}$. Тогда квадрат K_{s+1} является искомым. Действительно, множество значений функции f на наборах этого квадрата равно $A_s \cup A_{s+1}$. Значит, f на наборах этого квадрата принимает по крайней мере три попарно различных значения ($\varepsilon_1,\varepsilon_2$ и какое-то еще из A_{s+1}). Поэтому одно из этих значений на наборах квадрата K_{s+1} принимается ровно один раз, что и доказывает лемму.

Перейдем теперь непосредственно к доказательству теоремы 2.

Достаточность (*теорема* 2). Сначала построим все функции, принимающие ровно два значения. А затем по индукции из всех функций, принимающих не более чем $l-1$ значений, получим все функции, принимающие l значений, $l \leqslant k$.

Пусть $h(x_1,\dots,x_n)$ — существенная функция из F. По лемме о квадрате найдутся наборы, образующие квадрат, на котором неко-

торое значение, например ε, принимается ровно один раз. Пусть эти наборы имеют вид, представленный в левой части табл. 1; в правой части выписаны соответствующие им значения функции f. Тогда $\varepsilon \in \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$.

Таблица 1

x_1	x_2	x_3	...	x_n	$f(x_1, \ldots, x_n)$
α	β	α_3	...	α_n	ε
γ	β	α_3	...	α_n	ε_1
α	δ	α_3	...	α_n	ε_2
γ	δ	α_3	...	α_n	ε_3

Рассмотрим функцию $\varphi(x_1, x_2) = f(x_1, x_2, \alpha_3, \ldots, \alpha_n)$. Так как константы — функции от одной переменной, принимающие одно значение, то $\varphi \in [F]$. Рассмотрим также следующие функции:

$$\psi(x) = \begin{cases} 0, \text{ если } x = \varepsilon; \\ 1, \text{ если } x \neq \varepsilon, \end{cases}$$

$$\lambda_1(x) = \begin{cases} \alpha, \text{ если } x = 0; \\ \gamma, \text{ если } x \neq 0, \end{cases}$$

$$\lambda_2(x) = \begin{cases} \beta, \text{ если } x = 0; \\ \delta, \text{ если } x \neq 0. \end{cases}$$

Все эти функции принимают $2 \leqslant k - 1$ значения. Поэтому они принадлежат системе F. Положим $\omega(x_1, x_2) = \psi(\varphi(\lambda_1(x_1), \lambda_2(x_2)))$. Легко видеть, что на наборах из нулей и единиц функция ω ведет себя как дизъюнкция x_1 и x_2 (см. табл. 2). Обозначим ее через $\vee_{(0,1)}(x_1, x_2)$.

Таблица 2

x_1	x_2	$\omega(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	1

Кроме того, системе F принадлежат функции $j_\alpha(x),\ \alpha \in E_k$:

$$j_\alpha(x) = \begin{cases} 1, \text{ если } x = \alpha, \\ 0, \text{ если } x \neq \alpha, \end{cases}$$

поскольку они принимают только два значения.

Рассмотрим функцию $\&_{(0,1)}(x_1, x_2) = j_0(\omega(j_0(x_1), j_0(x_2)))$. Легко видеть, что на наборах из нулей и единиц она ведет себя как конъюнкция x_1 и x_2.

Эти функции позволяют получить все функции из P_k, принимающие любые два значения. Пусть $g(x_1, \ldots, x_n)$ — произвольная функция, принимающая только значения 0 и 1. Тогда

$$g(x_1, \ldots, x_n) =$$
$$= \bigvee_{(\sigma_1, \ldots, \sigma_n)} {}_{(0,1)}\, j_{\sigma_1}(x_1) \,\&_{(0,1)} \ldots \&_{(0,1)}\, j_{\sigma_n}(x_n) \&_{(0,1)}\, g(\sigma_1, \ldots, \sigma_n).$$

Доказательство этого равенства проводится прямой проверкой (как ранее для аналогичных представлений).

Пусть теперь $h(x_1, \ldots, x_n)$ — произвольная функция, принимающая только значения α и β $(\alpha \neq \beta)$. Рассмотрим функции

$$g(x_1, \ldots, x_n) = \begin{cases} 0, \text{ если } h(x_1, \ldots, x_n) = \alpha; \\ 1, \text{ если } h(x_1, \ldots, x_n) = \beta, \end{cases}$$

$$\mu(x) = \begin{cases} \alpha, \text{ если } x = 0; \\ \beta, \text{ если } x \neq 0. \end{cases}$$

Очевидно, что $\mu \in F$ и, как установлено выше, функция g принадлежит $[F]$. Кроме того, справедливо равенство $h(x_1, \ldots, x_n) = \mu(g(x_1, \ldots, x_n))$. Поэтому $h \in [F]$. Таким образом, все функции, принимающие произвольные два значения, выражаются формулами над F.

Построим теперь функции, принимающие l значений, из функций, принимающих не более $l - 1$ значений, $l - 1 < k$.

Сначала построим функции, принимающие некоторые l значений из множества E_k. По лемме о трех наборах найдутся такие наборы $\tilde{\beta}^1 = (\alpha, \alpha_2, \ldots, \alpha_n)$, $\tilde{\beta}^2 = (\beta, \alpha_2, \ldots, \alpha_n)$ и $\tilde{\beta}^3 = (\alpha, \gamma_2, \ldots, \gamma_n)$, что (существенная) функция $f(x_1, \ldots, x_n)$ на них принимает три различных значения. Пусть $f(\tilde{\beta}^1) = \varepsilon_1$, $f(\tilde{\beta}^2) = \varepsilon_2$, $f(\tilde{\beta}^3) = \varepsilon_3$. Обозначим компоненты этих наборов следующим образом:

$$\tilde{\beta}^1 = (\beta_1^1, \ldots, \beta_n^1), \quad \tilde{\beta}^2 = (\beta_1^2, \ldots, \beta_n^2), \quad \tilde{\beta}^3 = (\beta_1^3, \ldots, \beta_n^3).$$

Выберем произвольные различные значения $\varepsilon_4, \ldots, \varepsilon_l$ из множества $E_k \setminus \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$. Получим множество $I = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_l\}$, $|I| = l$. Так как функция f принимает все значения из множества

E_k, то найдутся такие наборы $\tilde{\beta}^i = (\beta_1^i, \ldots, \beta_n^i)$, что выполняются равенства $f(\tilde{\beta}^i) = \varepsilon_i$, $i = 4, \ldots, l$. Представим эти наборы в виде таблицы, в левой части которой по строкам выписаны сами наборы, а в правой части стоят соответствующие им значения функции f (см. табл. 3).

Таблица 3

x_1	x_2	$\ldots$	x_n	$f(x_1, \ldots, x_n)$
β_1^1	β_2^1	$\ldots$	β_n^1	ε_1
		$\ldots$		$\ldots$
β_1^i	β_2^i	$\ldots$	β_n^i	ε_i
		$\ldots$		$\ldots$
β_1^l	β_2^l	$\ldots$	β_n^l	ε_l

Из свойств наборов $\tilde{\beta}^1$, $\tilde{\beta}^2$ и $\tilde{\beta}^3$ следует, что в каждом столбце левой части таблицы находится не более $l-1$ разных значений из множества E_k.

Пусть $g(x_1, \ldots, x_m)$ — произвольная функция, принимающая l значений из множества I. Поставим в соответствие каждому набору $\tilde{\sigma} = (\sigma_1, \ldots, \sigma_m)$ значений переменных $x_1, \ldots, x_m$ число $i(\tilde{\sigma})$ из множества $\{1, 2, \ldots, l\}$, такое, что $g(\tilde{\sigma}) = \varepsilon_{i(\tilde{\sigma})}$. Определим функции $h_1(x_1, \ldots, x_m), \ldots, h_n(x_1, \ldots, x_m)$ следующим образом. Положим $h_j(\sigma_1, \ldots, \sigma_m) = \beta^j_{i(\tilde{\sigma})}$, $j = 1, \ldots, n$, где β_i^j — компоненты наборов $\tilde{\beta}^1, \ldots, \tilde{\beta}^l$. Отметим, что каждая из этих функций принимает не более чем $l-1$ значений. Поэтому по предположению $h_1, \ldots, h_n \in [F]$. Кроме того, выполняется равенство

$$g(x_1, \ldots, x_m) = f(h_1(x_1, \ldots, x_m), \ldots, h_n(x_1, \ldots, x_m)).$$

Поэтому $g \in [F]$.

Пусть теперь $h(x_1, \ldots, x_n)$ — произвольная функция, принимающая произвольные l, $l \leqslant k-1$, значений, например, из множества $\{\eta_1, \ldots, \eta_l\} \subset E_k$. Рассмотрим функцию $\eta(x)$, такую, что

$$\eta(x) = \begin{cases} \eta_i, \text{ если } x = \varepsilon_i,\ i = 1, \ldots, l; \\ \eta_1 \text{ в остальных случаях.} \end{cases}$$

Определим также функцию $g(x_1, \ldots, x_n)$ следующим образом: $g(x_1, \ldots, x_n) = \varepsilon_i$ тогда и только тогда, когда $h(x_1, \ldots, x_n) = \eta_i$,

$i = 1, \ldots, l$. Очевидно, что $h(x_1, \ldots, x_n) = \eta(g(x_1, \ldots, x_n))$. Функция η принимает $l \leqslant k-1$ значений. Поэтому по условию $\eta \in F$, а функция g принимает значения из множества $I = \{\varepsilon_1, \ldots, \varepsilon_l\}$. Поэтому $g \in [F]$. Таким образом, $h \in [F]$.

Отметим, что при $l = k$ последний этап не нужен, поскольку множество $I = E_k$. Это завершает доказательство теоремы 2.

Лекция № 6

В качестве примера рассмотрим следующее приложение критерия полноты, доказанного на предыдущей лекции.

Функция $f \in P_k$ называется *функцией Шеффера*, если $\{f\}$ — полная система в P_k.

Утверждение. *Функция $f(x_1, \ldots, x_n)$ из P_k, $k \geqslant 3$, является функцией Шеффера тогда и только тогда, когда f порождает все функции одной переменной, принимающие не более $k-1$ значений.*

Доказательство. Необходимость очевидна. Покажем, что f — существенная функция. Предположим, что f не принимает некоторое значение α из E_k. Тогда любая формула над $\{f\}$ не принимает значение α. А значит, константа α не выражается формулами над $\{f\}$, что противоречит условию. Поэтому функция f принимает все значения из E_k. Предположим, что f имеет не более одной существенной переменной. Поскольку f принимает все k значений, то она является перестановкой. Поэтому любая формула над $\{f\}$ реализует некоторую перестановку. А значит, из f нельзя получить никакую другую функцию одной переменной (например, константу). Таким образом, функция f принимает все k значений и существенно зависит по крайней мере от двух переменных, т. е. f — существенная функция. Поэтому в силу критерия Яблонского $\{f\}$ — полная система.

Коснемся теперь отличий k-значной логики от двузначной. Рассмотрим сначала вопрос о представлении функции полиномами. Докажем предварительно малую теорему Ферма.

Лемма 1. *Если k простое и $a \not\equiv 0 \pmod k$, то $a^{k-1} = 1 \pmod k$.*

Доказательство. Пусть $a \in E_k \setminus \{0\}$. Рассмотрим числа $b_1 = a\cdot 1, b_2 = a\cdot 2, \ldots, b_{k-1} = a\cdot(k-1)$. Предположим, что для некоторых i, j, $i \neq j$, выполняется равенство $b_i = b_j$. Тогда число $b_i - b_j = a(i-j)$ делится на k. Поскольку a и k — взаимно простые числа, то $i - j = 0$, т. е. $i = j$. Таким образом, $\{b_1, b_2, \ldots, b_{k-1}\} = E_k \setminus \{0\}$. Рассмотрим произведение $b_1 \cdot b_2 \cdot \ldots \cdot b_{k-1}$ этих чисел. Имеем

$$b_1 \cdot b_2 \cdot \ldots \cdot b_{k-1} = (a\cdot 1)(a\cdot 2)\ldots(a\cdot(k-1)) = 1\cdot 2\cdot\ldots\cdot(k-1).$$

Получаем равенство $a^{k-1}(k-1)! = (k-1)!$. Так как $(k-1)! \not\equiv 0 \pmod k$, то $a^{k-1} = 1 \pmod k$.

Теорема. *Система*

$$F = \{0, 1, \ldots, k-1,\ xy \pmod k,\ x+y \pmod k\}$$

полна в P_k тогда и только тогда, когда k — простое число.

Доказательство. Пусть k — простое число. Рассмотрим функцию $\varphi(x) = 1 - x^{k-1} \pmod k$. Тогда $\varphi(0) = 1$, а в силу леммы 1 $\varphi(x) = 0$ при всех $x \neq 0$. Поэтому $\varphi(x) = j_0(x)$. Легко видеть, что выполняется равенство $j_i(x) = j_0(x - i)$, $i = 1, \ldots, k-1$. Таким образом, функции $j_0, j_1, \ldots, j_{k-1}$ выражаются формулами над F. Кроме того, для любой функции $f(x_1, \ldots, x_n)$ из P_k имеет место представление

$$f(x_1, \ldots, x_n) = \sum_{(\sigma_1, \ldots, \sigma_n)} j_{\sigma_1}(x_1)\ldots j_{\sigma_n}(x_n) f(\sigma_1, \ldots, \sigma_n) \pmod k.$$

Поэтому F — полная система в P_k.

Предположим, что $k = k_1 k_2$, $1 < k_1, k_2 < k$. Покажем, что функция $j_0(x)$ не представляется в виде полинома. Предположим противное. Пусть выполняется равенство

$$j_0(x) = c_0 + c_1 x + \ldots + c_t x^t,$$

где $c_0, c_2, \ldots, c_t \in E_k$, $1 \leqslant t < k-1$. Так как $j_0(0) = 1$, то $c_0 = 1$. При $x = k_1$ получаем

$$0 = 1 + c_1 k_1 + \ldots + c_t k_1^t \pmod k.$$

Умножив обе части равенства на k_2, получаем $0 = k_2 \pmod k$, что противоречит предположению. Таким образом, при составном k система F не является полной в P_k.

Напомним, что каждый замкнутый класс в P_2 (из счетного семейства замкнутых классов) имеет конечный базис. В P_k при $k \geqslant 3$ это утверждение неверно, т. е. существуют замкнутые классы, которые не имеют базиса. Пример такого класса впервые был построен Ю. И. Яновым.

Утверждение. *Для любого k, $k \geqslant 3$, в P_k существуют замкнутые классы, не имеющие базиса.*

Доказательство. Рассмотрим следующую систему функций: $F = \{f_0, f_1, \ldots, f_n, \ldots\}$, где $f_0 = 0$,

$$f_n(x_1, \ldots, x_n) = \begin{cases} 1 & \text{при } x_1 = x_2 = \ldots = x_n = 2; \\ 0 & \text{в остальных случаях} \end{cases}$$

для каждого $n \geqslant 1$. Положим $M_k = [F]$. Очевидно, что для всех $m \geqslant 1$ и любых $\alpha_1, \ldots, \alpha_m \in E_k$ выполняется неравенство $f_m(\alpha_1, \ldots, \alpha_m) \neq 2$. Поэтому каждая формула Φ над F, которая имеет вид

$$\Phi = f_n(A_1, \ldots, A_n)$$

и в которой среди $A_1, \ldots, A_n$, $n \geqslant 1$, найдется по крайней мере одно выражение, отличное от символа переменной, реализует константу $0 = f_0$. Следовательно, каждая функция из M_k получается из функций системы F некоторой подстановкой переменных.

Покажем, что класс M_k не имеет базиса. Предположим противное. Пусть множество функций G — базис класса M_k. Тогда если G содержит по крайней мере две функции $f_n(x_{i_1}, \ldots, x_{i_n})$ и $f_m(x_{j_1}, \ldots, x_{j_m})$, у которых число существенных переменных n и m соответственно, $n \geqslant m$, то функция f_m получается из f_n подстановкой (в частности, отождествлением) переменных. Если же $G = \{f_m\}$, $m \geqslant 1$, то в силу установленных выше свойств никакая функция f_n из F при $n > m$ не может выражаться формулой над $\{f_m\}$. Таким образом, множество M_k не имеет базиса.

Приведем теперь пример замкнутого класса в P_k со счетным базисом, $k \geqslant 3$. Определим для всех $n \geqslant 2$ множество наборов R_n следующего вида:

$$R_n = \{(\alpha_1, \ldots, \alpha_n) \,|\, \alpha_j = 1,$$
$$\alpha_1 = \ldots = \alpha_{j-1} = \alpha_{j+1} = \ldots = \alpha_n = 2,\ j = 1, 2, \ldots, n\}.$$

Каждый набор из R_n имеет ровно одну единичную компоненту, все остальные компоненты этого набора равны 2. Рассмотрим систему функций $F = \{\Psi_2, \Psi_3, \ldots\}$ из P_k, такую, что

$$\Psi_n(x_1, \ldots, x_n) = \begin{cases} 1, & \text{если } (x_1, \ldots, x_n) \in R_n; \\ 0 & \text{в противном случае,} \end{cases}$$

$n \geqslant 2$. Легко видеть, что при всех $n \geqslant 2$ функции Ψ_n существенно зависят от всех переменных и принимают значения из множества $\{0, 1\}$. Положим $G_k = [F]$. Покажем, что G_k является искомым классом.

Утверждение. *Для любого $n \geqslant 2$ функция Ψ_n не выражается формулой над системой*

$$F \setminus \{\Psi_n\}.$$

Доказательство. Предположим противное. Пусть Φ — формула над $F \setminus \{\Psi_n\}$, реализующая функцию Ψ_n. Пусть она имеет вид

$$\Phi = \Psi_m(A_1, \ldots, A_m),$$

где $A_1, \ldots, A_m$ — либо формулы над $F \setminus \{\Psi_n\}$, либо переменные, $m \geqslant 2$, $m \neq n$. Без ограничения общности будем считать, что формула Φ содержит переменные только из множества $\{x_1 \ldots, x_n\}$. Рассмотрим три возможных случая.

1. Среди $A_1, \ldots, A_m$ по крайней мере две формулы отличны от символов переменных, например A_i и A_j, $i \neq j$. Тогда для любого набора $\alpha = (\alpha_1, \ldots, \alpha_n)$ значения $A_i(\alpha)$ и $A_j(\alpha)$ этих формул на наборе α принадлежат множеству $\{0, 1\}$. Поэтому набор $(A_1(\alpha), \ldots, A_m(\alpha))$ не принадлежит множеству R_m. А значит, $\Phi(\alpha) = \Psi_m(A_1(\alpha), \ldots, A_m(\alpha)) = 0$, что противоречит исходному предположению, поскольку функция Ψ_n не является константой.

2. Среди $A_1, \ldots, A_m$ есть ровно одна формула A_i, которая отлична от символа переменной. Так как $m \geqslant 2$, то среди $A_1, \ldots, A_m$ найдется по крайней мере один символ переменной; пусть, например, A_j — это символ переменной x_s, $1 \leqslant s \leqslant n$. Рассмотрим набор $\alpha = (\alpha_1, \ldots.\alpha_n)$, такой, что

$$\alpha_s = 1,\ \alpha_1 = \ldots = \alpha_{s-1} = \alpha_{s+1} = \ldots = \alpha_n = 2.$$

С одной стороны, набор α принадлежит множеству R_n и поэтому $\Psi_n(\alpha) = 1$. С другой стороны, так как $A_i(\alpha) \in \{0, 1\}$, то набор $(A_1(\alpha), \ldots, A_m(\alpha))$ не принадлежит множеству R_m и поэтому $\Phi(\alpha) = \Psi_m(A_1(\alpha), \ldots, A_m(\alpha)) = 0$, что противоречит предположению.

3. Все $A_1, \ldots, A_m$ являются символами переменных, например $x_{i_1}, \ldots, x_{i_m}$ соответственно; $x_{i_1}, \ldots, x_{i_m} \in \{x_1, \ldots, x_n\}$. Имеем

$$\Psi_n(x_1, \ldots, x_n) = \Psi_m(x_{i_1}, \ldots, x_{i_m}),$$

где $m \neq n$. Так как функция Ψ_n существенно зависит от всех переменных, то $m \geqslant n$. А поскольку $m \neq n$, то $m > n$. Поэтому среди переменных $x_{i_1}, \ldots, x_{i_m}$ есть две одинаковые. Пусть, например, $x_{i_1} = x_{i_2} = x_l$, $1 \leqslant l \leqslant n$. Рассмотрим набор $\alpha = (\alpha_1, \ldots, \alpha_n)$, такой, что

$$\alpha_l = 1,\ \alpha_1 = \cdots = \alpha_{l-1} = \alpha_{l+1} = \cdots = \alpha_n = 2.$$

Очевидно, что $\alpha \in R_n$, и поэтому $\Psi_n(\alpha) = 1$. С другой стороны, $\Phi(\alpha) = 0$, поскольку набор $(A_1(\alpha), \dots, A_m(\alpha))$ не принадлежит множеству R_m. Полученное противоречие доказывает утверждение.

Следствие. *Система F является базисом класса $G_k = [F]$.*

Получим еще одно важное следствие из предыдущего утверждения.

Утверждение. *Для любого $k \geqslant 3$ мощность семейства замкнутых классов k-значной логики равна континууму.*

Доказательство. Определим сначала на основе системы $F = \{\Psi_2, \Psi_3, \dots\}$ континуальное семейство замкнутых классов. Для каждого множества индексов $A = \{i_1, i_2, \dots\}$, такого, что $2 \leqslant i_1 < i_2, \dots$, построим систему

$$F(A) = \bigcup_{i \in A} \{\Psi_i\};$$

$F(A)$ является подсистемой системы F. Положим $G_A = [F(A)]$. Рассмотрим теперь два различных множества индексов A и B. Тогда найдется n, такое, что $n \in A$ и $n \notin B$. Поэтому $\Psi_n \in G(A)$, но в силу предыдущего утверждения $\Psi_n \notin G(B)$, т. е. $G(A) \neq G(B)$. Значит, различные множества индексов определяют разные замкнутые классы. Поскольку семейство различных подмножеств множества $\{2, 3, \dots\}$ имеет континуальную мощность, то класс G_k содержит континуальное семейство замкнутых классов.

С другой стороны, P_k содержит счетное множество функций, а каждый замкнутый класс является подмножеством этого множества. Поэтому мощность семейства замкнутых классов в P_k не превышает мощности семейства всех подмножеств множества P_k.

Приведенные выше утверждения говорят о принципиальных отличиях k-значных логик от двузначной.

Сформулируем теперь аналог теоремы о полноте для P_2, основанный на распознавании некоторых свойств функций. Пусть A — некоторое множество функций из P_k, удовлетворяющее следующим условиям:

1) каждая функция из A зависит от одного и того же набора переменных $y_1, \dots, y_p$, $p \geqslant 1$;
2) функции $g_i(y_1, \dots, y_p) = y_i$, $i = 1, \dots, p$, содержатся в A.

Говорят, что функция $f(x_1, \dots, x_n)$ *сохраняет множество* A, если для любых функций $h_{i_1}(y_1, \dots, y_p), \dots, h_{i_n}(y_1, \dots, y_p)$ из A функция

$$f(h_{i_1}(y_1, \dots, y_p), \dots, h_{i_n}(y_1, \dots, y_p))$$

принадлежит множеству A. Обозначим через M_A множество всех функций из P_k, сохраняющих множество A.

Лемма 2. *Пусть A — произвольное множество функций из P_k, удовлетворяющее условиям 1 и 2. Тогда M_A — замкнутый класс.*

Доказательство. Очевидно, что M_A содержит тождественную функцию. Поэтому достаточно показать, что если $f_0(z_1, \dots, z_m) \in M_A$ и $f_1, \dots, f_m \in M_A$, то и $f = f_0(f_1, \dots, f_m)$ принадлежит множеству M_A. Будем считать, что у всех функций $f_1, \dots, f_m$ одинаковый набор переменных $\tilde{x} = (x_1, \dots, x_n)$, т. е.

$$f(\tilde{x}) = f_0(f_1(\tilde{x}), \dots, f_m(\tilde{x})).$$

Возьмем произвольный набор функций $h_{i_1}(y_1, \dots, y_p), \dots, h_{i_n}(y_1, \dots, y_p)$ из A. Так как функции $f_1, \dots, f_m$ сохраняют множество A, то функции $H_1(y_1, \dots, y_p) = f_1(h_{i_1}, \dots, h_{i_n})$, …, $H_m(y_1, \dots, y_p) = f_m(h_{i_1}, \dots, h_{i_n})$ принадлежат A. Поэтому и функция

$$f(h_{i_1}, \dots, h_{i_n}) = f_0(f_1(h_{i_1}, \dots, h_{i_n}), \dots, f_m(h_{i_1}, \dots, h_{i_n})) = \\ = f_0(H_1, \dots, H_m)$$

принадлежит A.

Пусть F — произвольное множество функций из P_k, а $\tilde{x}$ — набор переменных $x_1, \dots, x_p$, $p \geqslant 1$. Обозначим через $F_{\tilde{x}}$ множество функций из F, зависящих от переменных $x_1, \dots, x_p$.

Лемма 3. *Пусть A — произвольное множество функций из P_k, удовлетворяющее условиям 1 и 2, такое, что $[A]_{\tilde{x}} = A$, где $\tilde{x} = (x_1, \dots, x_p)$. Тогда $(M_A)_{\tilde{x}} = A$.*

Доказательство. Пусть $f(x_1, \dots, x_p)$ — произвольная функция из A. Рассмотрим произвольные функции $h_{i_1}, \dots, h_{i_p}$ из A. Тогда функция $g = f(h_{i_1}, \dots, h_{i_p})$ принадлежит $[A]$ и зависит только от переменных $x_1, \dots, x_p$. Поэтому $g \in [A]_{\tilde{x}}$. Но по условию $[A]_{\tilde{x}} = A$. Поэтому функция f сохраняет множество A, т. е. $f \in (M_A)_{\tilde{x}}$.

С другой стороны, если функция f принадлежит $(M_A)_{\tilde{x}}$, то f сохраняет множество A и зависит только от переменных $x_1, \ldots, x_p$. Рассмотрим функцию

$$F(x_1, \ldots, x_p) = f(g_1(x_1, \ldots, x_p), \ldots, g_p(x_1, \ldots, x_p)).$$

Так как функции $f, g_1, \ldots, g_p$ сохраняют множество A, то $F \in A$. Но для всех $i = 1, \ldots, p$ выполняется равенство $g_i(x_1, \ldots, x_p) = x_i$. Поэтому $F(x_1, \ldots, x_p) = f(x_1, \ldots, x_p)$, т. е. $f \in A$.

Докажем теперь теорему А. В. Кузнецова.

Теорема (о функциональной полноте). *Существуют замкнутые классы $M_1, \ldots, M_s$, такие, что ни один из них не содержится ни в одном из остальных и произвольная система F из P_k полна тогда и только тогда, когда F целиком не содержится ни в одном из классов $M_1, \ldots, M_s$.*

Доказательство. Построим сначала систему классов. Пусть $A_1, \ldots, A_l$ — система всех собственных подмножеств множества $(P_k)_{x_1x_2}$ (множества всех функций из P_k от переменных x_1 и x_2), таких, что для всех $i = 1, \ldots, l$ выполняются следующие условия:

1) функции $g_1(x_1, x_2) = x_1$ и $g_2(x_1, x_2) = x_2$ содержатся в A_i;
2) $[A_i]_{x_1x_2} = A_i$.

Указанная система может быть построена путем перебора всех собственных подмножеств множества $P_k(2)$. Поскольку $|P_k(2)| = k^{k^2}$, то число таких подмножеств не превышает $2^{k^{k^2}}$.

Положим $H_i = M_{A_i}$. Из лемм 2 и 3 следует, что для всех значений $i = 1, \ldots, l$ множество H_i — замкнутый класс, такой, что $[H_i]_{x_1x_2} = A_i$. Удаляя из построенной системы $H_1, \ldots, H_l$ те классы, которые содержатся в каком-либо из остальных, получаем систему $M_1, \ldots, M_s$ классов, таких, что $M_i \neq P_k$, $M_i \not\subseteq M_j$ при всех $i, j = 1, \ldots, s$, $i \neq j$.

Покажем теперь, что построенная система классов является искомой. Пусть F — произвольная система функций из P_k. Очевидно, что если при некотором i , $1 \leqslant i \leqslant s$, выполняется включение $F \subseteq M_i$, то $[F] \subseteq M_i \neq P_k$. То есть F — неполная система.

Пусть F целиком не содержится ни в одном из классов $M_1, \ldots, M_s$. Положим

$$F_1 = F \cup \{g_1(x_1, x_2), g_2(x_1, x_2)\}.$$

Очевидно, что F — полная система тогда и только тогда, когда F_1 — полная система.

Положим $B = [F_1]_{x_1x_2}$. Покажем, что B содержит все функции из P_k от переменных x_1 и x_2. Предположим, что это не так. Тогда $B \neq (P_k)_{x_1x_2}$. Поскольку B содержит функции $g_1(x_1, x_2) = x_1$ и $g_2(x_1, x_2) = x_2$, а $[B]_{x_1x_2} = B$, то найдется такое i, $1 \leqslant i \leqslant l$, что выполняется равенство $B = A_i$. Так как каждая функция из F_1 сохраняет множество $B = [F_1]_{x_1x_2}$, то $F_1 \subseteq H_i = M_{A_i}$. Поэтому найдется такое j, $1 \leqslant j \leqslant s$, что $F_1 \subseteq H_i \subseteq M_j$. Так как $F \subseteq F_1$, то $F \subseteq M_j$, что противоречит условию. Итак, $B = (P_k)_{x_1x_2}$. Поэтому F_1 содержит функцию $V_k(x_1, x_2) = \max(x_1, x_2) + 1$. А значит, и система F содержит функцию $V_k(x_1, x_2)$. Таким образом, F — полная система.

ЛОГИЧЕСКИЕ СХЕМЫ

Лекция № 7

Графы. Пусть имеются два конечных (или счетных) множества $V = \{v_1, \ldots, v_n\}$ и $E = \{e_1, \ldots, e_k\}$ и каждому элементу e_i из E поставлена в соответствие неупорядоченная пара (v_{i_1}, v_{i_2}) элементов из V (при этом допускаются пары (v, v) из одинаковых элементов; разным элементам из E может соответствовать одна и та же пара элементов из V; некоторые элементы из V могут не входить ни в одну из пар, соответствующих элементам из E). Полученный объект называется *графом*. Элементы множества V называются *вершинами* графа, элементы множества E — *ребрами* графа. Иными словами, граф можно рассматривать как совокупность трех упорядоченных объектов (V, E, ρ) — множества вершин V, множества ребер E и отображения $\rho : E \to V_1 \cup V_2$ множества E в множество всех неупорядоченных пар элементов из V (V_1 и V_2 — множества всех одноэлементных и двухэлементных подмножеств множества V соответственно). Если $\rho(e) \in V_1$, то ребро e называется *петлей*; если $\rho(e_1) = \rho(e_2)$, то ребра e_1 и e_2 называются *кратными* или *параллельными*. Вершины v_{i_1} и v_{i_2} из V, которые образуют пару (v_{i_1}, v_{i_2}), соответствующую ребру $e_i \in E$, называются *концами* этого ребра; при этом говорят, что ребро e_i соединяет v_{i_1} и v_{i_2}. Вершины v_{i_1} и v_{i_2} называются также *смежными*, а про каждую из них говорят, что она *инцидентна* ребру e_i. Число ребер, инцидентных вершине v, называется *степенью* вершины v и обозначается через $d(v)$. Если $d(v) = 0$, то вершина v называется *изолированной*; если $d(v) = 1$, то v называется *висячей* (или *концевой*).

В дальнейшем мы часто будем изображать ребро с концами v_{i_1} и v_{i_2} символом (v_{i_1}, v_{i_2}). Это будет делаться в тех случаях, когда из текста будет ясно, о каком именно ребре идет речь, или когда это несущественно.

Подграфом графа $G = (V, E, \rho)$ называется такой граф (V', E', ρ'), у которого все вершины и ребра принадлежат G, т. е. $V' \subseteq V$, $E' \subseteq E$ и $\rho' = \rho|_{E'}$, где $\rho|_{E'}$ — это ограничение ρ на E', т. е. для всех e из E' выполняется равенство $\rho'(e) = \rho(e)$; при этом необходимо, чтобы $\rho(e) \in V_1' \cup V_2'$ для любого e из E'.

Каждому графу сопоставим некоторую геометрическую фигуру (*геометрическая реализация графа*) следующим образом. Выберем на плоскости (или в пространстве) n точек и сопоставим их вершинам графа $v_1, \ldots, v_n$. Эти точки будем обозначать символами $v_1, \ldots, v_n$. Затем для каждого ребра (v_{i_1}, v_{i_2}) графа выберем отрезок (вообще говоря, криволинейный), соединяющий точки v_{i_1} и v_{i_2}. Геометрическая реализация называется *правильной*, если отрезки (или кривые) не имеют попарно общих точек, кроме общих инцидентных вершин.

Пример. Пусть $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$. Эти два множества вместе с соответствием ρ: $e_1 \to (1, 1)$, $e_2 \to (1, 2)$, $e_3 \to (1, 2)$, $e_4 \to (1, 3)$, $e_5 \to (1, 4)$, $e_6 \to (2, 3)$, $e_7 \to (2, 5)$ образуют граф G. Его правильная (плоская) геометрическая реализация изображена на рис. 1. У этого графа ребро e_1 — петля; ребра e_2 и e_3 кратные; вершины 1 и 4 смежные, а вершины 1 и 5 — нет; вершина 1 и ребро e_2 инцидентны; $d(1) = 5$; вершина 6 изолированная; вершины 4 и 5 концевые.

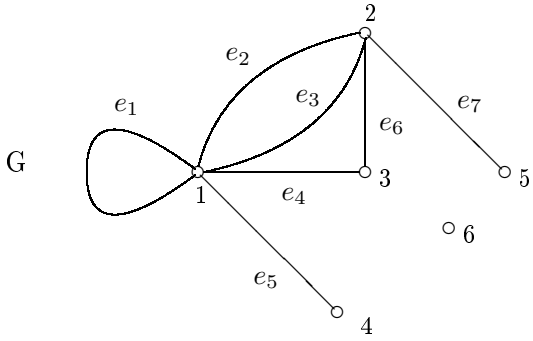


Рис. 1

Утверждение 1. *В трехмерном евклидовом пространстве для любого конечного графа имеется правильная геометрическая реализация.*

Доказательство. Пусть $g = (V, E, \rho)$ — произвольный конечный граф, $V = \{v_1, \ldots, v_n\}$, $E = \{e_1, \ldots, e_q\}$. Выберем в R^3 некоторую прямую I. Отметим на этой прямой n точек $1, 2, \ldots, n$ и сопоставим их вершинам графа $v_1, \ldots, v_n$. Проведем в R^3 q плоскостей $\pi_1, \ldots, \pi_q$, проходящих через прямую I, и сопоставим их ребрам графа $e_1, \ldots, e_q$. Затем в каждой плоскости π_i проведем кривую, которая соединяет концы v_{i_1} и v_{i_2} ребра $e_i = (v_{i_1}, v_{i_2})$ и не имеет других пересечений с прямой I (см. рис. 2). Очевидно, что полученная геометрическая реализация графа G является правильной.

Легко видеть, что эта конструкция легко обобщается на случай, когда граф G имеет счетное множество вершин и ребер.

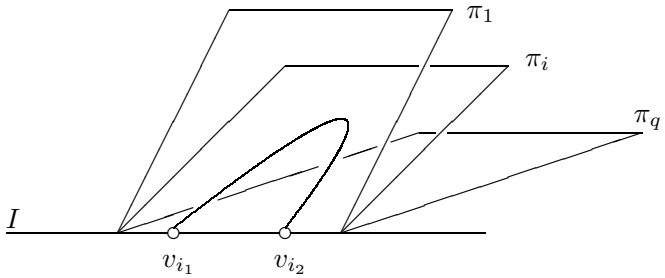


Рис. 2

Два графа $G_1 = \{V_1, E_1, \rho_1\}$ и $G_2 = \{V_2, E_2, \rho_2\}$ называются *изоморфными* (обозначение $G_1 \cong G_2$), если существуют взаимно однозначные соответствия $\varphi : V_1 \to V_2$ и $\psi : E_1 \to E_2$, такие, что для любого ребра $e = (v, w)$ из E_1 выполняется равенство $\psi(e) = (\varphi(v), \varphi(w))$.

Пусть $G = \{V, E, \rho\}$ — некоторый граф, $V = V_1 \cup \{v_1, v_2\}$, $v_1, v_2 \notin V_1$, $E = E_1 \cup \{e\}$, $e \notin E_1$, $e = (v_1, v_2)$, т. е. $\rho(e) = (v_1, v_2)$. Пусть $w \notin V$, $e_1, e_2 \notin E$. Определим граф $G_2 = (V_2, E_2, \rho_2)$ следующим образом. Положим $V_2 = V \cup \{w\}$, $E_2 = E_1 \cup \{e_1, e_2\}$, $\rho_2(e_1) = (v_1, w)$, $\rho_2(e_2) = (v_2, w)$, $\rho_2(x) = \rho(x)$ для всех $x \in E_1$. Переход от графа G к графу G_2 называется *разбиением ребра* e в графе G.

Два графа называются *гомеоморфными*, если их можно получить из одного графа с помощью последовательности разбиений ребер.

Примеры.

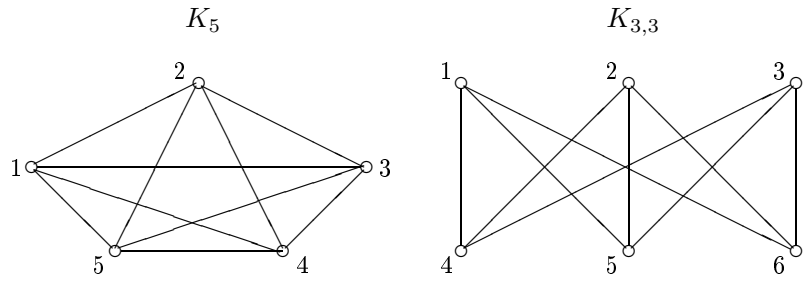
1. Граф K_5 — полный граф на пяти вершинах, каждые две из которых соединены ровно одним ребром (см. рис. 3, *а*).

2. Граф $K_{3,3} = (V, E, \rho)$, где

$$V = \{1, 2, 3, 4, 5, 6\}, \quad E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\},$$

$e_1 = (1, 4)$, $e_2 = (1, 5)$, $e_3 = (1, 6)$, $e_4 = (2, 4)$, $e_5 = (2, 5)$, $e_6 = (2, 6)$, $e_7 = (3, 4)$, $e_8 = (3, 5)$, $e_9 = (3, 6)$ (см. рис. 3, *б*).

Известно, что графы $K_{3,3}$ и K_5 не имеют правильной плоской реализации. В конце двадцатых годов прошлого века Л. С. Понтрягиным, а позже польским математиком К. Куратовским было установлено[1)], что граф допускает плоскую реализацию тогда и только тогда, когда он не содержит подграфа, гомеоморфного K_5 или $K_{3,3}$ (доказательства этой теоремы в этом курсе не будет).



а *б*

Рис. 3

Последовательность ребер $(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots, (v_{i_{k-1}}, v_{i_k})$ будем называть *путем*, соединяющим вершины v_{i_1} и v_{i_k}. Если все вершины $v_{i_1}, \ldots, v_{i_k}$ различны, то этот путь будем называть *цепью*. *Циклом* называется путь, у которого первая и последняя вершины совпадают (т. е. $v_{i_1} = v_{i_k}$); *простым циклом* называется цикл, у которого все вершины $v_{i_1}, \ldots, v_{i_{k-1}}$ различны и все ребра различны. Граф G называется *связным*, если любая пара его вершин соединена путем. *Деревом* называется связный граф без простых циклов.

Утверждение 2. *В каждом конечном связном графе можно выделить подграф, который содержит все вершины исходного графа и является деревом.*

Доказательство. Пусть G — произвольный связный конечный граф. Если G содержит простой цикл, то возьмем произвольное ребро этого цикла и удалим его из множества ребер графа G. В результате получим подграф G_1 графа G. Очевидно, что G_1 является связным и содержит все вершины графа G. Повторяя эту процедуру необходимое число раз, получим искомое дерево.

Граф $G = (V, E, \rho)$ называется *ориентированным*, если соот-

[1)] Доказательство см., например, в книге: *Харари Ф.* Теория графов. М.: Мир, 1973.

ветствие ρ имеет вид $\rho : E \to V \times V$. В этом случае говорят, что каждому ребру $e \in E$ приписано направление; если $\rho(e) = (v, w)$, то ребро e *выходит* из вершины v и *входит* в вершину w. *Ориентированным циклом* называется конечная последовательность ориентированных ребер $(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots, (v_{i_k}, v_{i_1})$.

Утверждение 3. *В любом конечном ориентированном графе без ориентированных циклов найдется вершина, из которой не выходит ни одно ребро.*

Доказательство. Предположим противное. Пусть из каждой вершины графа G выходит по крайней мере одно ребро. Возьмем произвольную вершину v_{i_1} графа G и произвольное ребро, которое выходит из v_{i_1}. Пусть оно имеет вид (v_{i_1}, v_{i_2}). Затем возьмем произвольное ребро графа G, выходящее из вершины v_{i_2} и так далее. В результате после k шагов, $k \geqslant |V|$ (где V — множество вершин графа G), мы получим ориентированный путь

$$(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}), \ldots, (v_{i_k}, v_{i_{k+1}}),$$

который в силу конечности графа G будет содержать ориентированный цикл. Полученное противоречие доказывает утверждение.

Следует отметить, что для бесконечных графов аналогичное утверждение неверно. Действительно, граф, множество вершин которого совпадает с множеством Z целых чисел, а множество ребер состоит из всех пар $(i, i+1)$ идущих подряд целых чисел, не имеет ориентированных циклов, но из каждой его вершины выходит ребро.

Лемма 1 (*о нумерации вершин*). *В любом конечном ориентированном графе без ориентированных циклов можно занумеровать вершины первыми натуральными числами так, что каждое ребро будет направлено от вершины с меньшим номером в вершину с бóльшим номером.*

Доказательство. Будем доказывать лемму индукцией по числу p вершин графа G. При $p = 1$ утверждение очевидно. Пусть $p > 1$. Предположим, что утверждение справедливо для всех конечных ориентированных графов без ориентированных циклов с числом вершин $p' < p$. Рассмотрим произвольный граф G с p вершинами, удовлетворяющий условиям леммы. Из утверждения 3 следует, что найдется вершина v, из которой не выходит ни одно ребро. Удалим из множества вершин графа G вершину v, а из множества ребер

графа G — все ребра, входящие в вершину v. Получим подграф G_1 графа G. Очевидно, что он удовлетворяет условиям леммы и имеет $p' = p - 1$ вершин. По предположению индукции граф G_1 допускает искомую нумерацию числами $1, 2, \ldots, p-1$. Вернемся теперь к графу G. Присвоим вершине v номер p. Таким образом, получили нумерацию вершин графа G числами $1, 2, \ldots, p$. Покажем, что она обладает требуемым свойством. Рассмотрим произвольное ребро $e = (v_{i_1}, v_{i_2})$ графа G. Если ребро e входит в вершину v (т. е. $v_{i_2} = v$), то v_{i_1} — вершина графа G_1 и поэтому ее номер не превышает $p - 1$. Если же $v_{i_2} \neq v$, то e является ребром графа G_1 и поэтому по предположению индукции направлено от вершины с меньшим номером в вершину с бóльшим номером.

Схемы из функциональных элементов. *Схема из функциональных элементов* (СФЭ) — это ориентированный граф без ориентированных циклов, в каждую вершину которого входит не более двух ребер и вершинам которого приписаны символы в соответствии со следующими правилами:

a) если в вершину не входят ребра, то этой вершине приписывается символ некоторой переменной;

b) если в вершину входит одно ребро, то этой вершине приписывается символ $\overline{\ }$ (отрицание);

c) если в вершину входят два ребра, то этой вершине приписывается либо символ & (конъюнкция), либо символ $\vee$ (дизъюнкция);

d) кроме того, некоторым вершинам приписывается символ $*$.

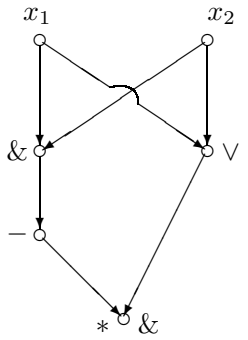


Рис. 4

Пример схемы из функциональных элементов изображен на рис. 4.

Каждой вершине СФЭ можно поставить в соответствие некоторую функцию алгебры логики по следующим правилам:

1) если вершине была приписана переменная, то ей ставится в соответствие функция, равная этой переменной;

2) если данной вершине v был приписан знак отрицания, а вершине, из которой выходит входящее в вершину v ребро, уже поставлена в соответствие функция φ, то вершине v ставится в соответствие функция $\overline{\varphi}$;

3) если данной вершине v был приписан знак $\&$ (соответственно $\vee$), а вершинам, из которых выходят входящие в вершину v ребра, уже поставлены в соответствие функции φ_1 и φ_2, то данной вершине ставится в соответствие функция $\varphi_1\&\varphi_2$ (соответственно $\varphi_1 \vee \varphi_2$).

Этот процесс в конце концов поставит каждой вершине графа некоторую функцию алгебры логики от переменных, приписанных вершинам схемы (это следует, например, из возможности занумеровать вершины в соответствии с леммой о нумерации вершин); при этом ставить в соответствие вершинам функции следует в порядке их нумерации. Соответствующий пример изображен на рис. 5.

Функции, поставленные в соответствие вершинам, помеченным символом $*$, по определению *реализуются* этой схемой.

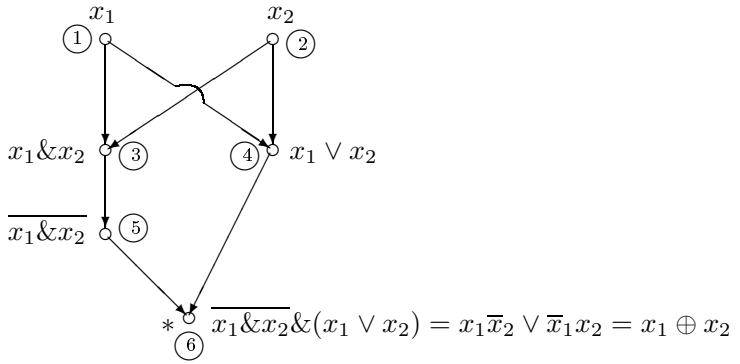


Рис. 5

Вершины, которым приписаны переменные, называются *входами* схемы; вершины, которым приписаны символы $\&, \vee, \overline{\ }$, называются *элементами* (*функциональными элементами*); вершины, которым приписан символ $*$, — *выходами* схемы. Обычно элементы в схеме изображаются в виде геометрических фигур (например, треугольников), внутри которых находятся символы, приписанные этим элементам (рис. 6).

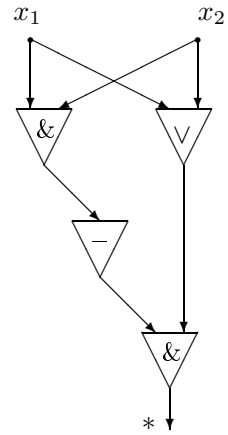


Рис. 6

С каждой СФЭ, в которой вершины занумерованы правильным образом, можно связать систему уравнений, описывающую вычисление функций, сопоставленных вершинам. Для нашего примера эта система выглядит следующим образом:

$$y_1 = x_1, \quad y_2 = x_2,$$
$$y_3 = y_1\&y_2, \quad y_4 = y_1 \vee y_2,$$
$$y_5 = \overline{y}_3, \quad y_6 = y_4\&y_5.$$

Имеется физическая интерпретация СФЭ, в которой они рассматриваются как математические модели реальных электронных схем: если на вход подается набор значений (наличие тока соответствует единице, отсутствие — нулю), то на выходе получается значение функции на этом наборе.

Сложностью схемы S будем называть число ее элементов (обозначение $L(S)$). Схема на рис. 6 имеет сложность 4. Пусть f — функция алгебры логики. Положим

$$L(f) = \min_{S \text{ реализует } f} L(S).$$

Величина $L(f)$ называется *сложностью* функции f (в классе СФЭ). Определим теперь следующую функцию:

$$L(n) = \max_{f(x_1,\ldots,x_n)\in P_2} L(f(x_1,\ldots,x_n)).$$

$L(n)$ называется *функцией Шеннона*. Другими словами, $L(n)$ есть наименьшее число элементов, достаточное для реализации любой булевой функции от переменных $x_1,\ldots,x_n$.

Опишем простейший метод синтеза, основанный на моделировании совершенной дизъюнктивной нормальной формы.

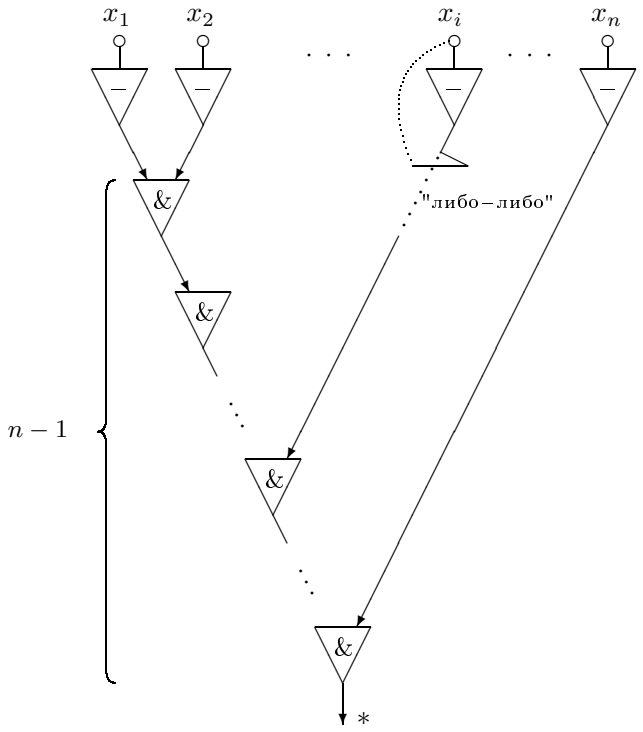


Рис. 7

Лемма 2. *Для любой конъюнкции* $x_1^{\sigma_1}\ldots x_n^{\sigma_n}$

$$L(x_1^{\sigma_1}\ldots x_n^{\sigma_n}) \leqslant 2n-1.$$

Доказательство. Схема может быть построена из n элементов отрицания, присоединенных к входам, и цепочки из элементов конъюнкции, имеющих n "свободных" входов. Каждый (i-й) вход этой цепочки присоединяется к входу схемы, если i-й множитель равен x_i, или к выходу i-го элемента отрицания, если i-й множитель равен $\overline{x}_i$ (рис. 7).

Очевидно, что сложность построенной схемы равна $2n-1$. Поэтому $L(x_1^{\sigma_1}\ldots x_n^{\sigma_n}) \leqslant 2n-1$.

Теорема 1. *Имеет место неравенство*

$$L(n) \leqslant n2^{n+1}.$$

Доказательство. Пусть $f(x_1,\ldots,x_n)$ — произвольная булева функция. Если $f \neq 0$, то f может быть задана совершенной дизъюнктивной нормальной формой

$$f(x_1,\ldots,x_n) = K_1 \vee K_2 \vee \cdots \vee K_s,$$

где $s \leqslant 2^n$ и каждая конъюнкция имеет вид

$$K_j = x_1^{\sigma_{j_1}} x_2^{\sigma_{j_2}} \ldots x_n^{\sigma_{j_n}}.$$

Схема S для f состоит из конъюнкций K_j (каждая из них в соответствии с леммой 2 имеет сложность не более $2n-1$) и цепочки из $s-1$ элемента дизъюнкции с s свободными входами; свободные входы этой цепочки присоединяются к выходам схем для конъюнкций K_j (рис. 8). Имеем

$$L(S) \leqslant s(2n-1)+s-1 < s(2n-1)+s = 2ns \leqslant n2^{n+1}.$$

Если $f = 0$, то схема строится в соответствии с представлением $0 = x_1\&\overline{x}_1$, т. е. $L(0) \leqslant 2$.

Таким образом, для любой функции $f(x_1,\ldots,x_n)$ выполняется неравенство

$$L(f(x_1,\ldots,x_n)) \leqslant n2^{n+1}.$$

Поэтому

$$L(n) \leqslant n2^{n+1}.$$

Теорема доказана.

Данный метод реализации функций схемами имеет существенный недостаток: каждая конъюнкция реализуется отдельно. Поэтому возникает много дублирующих друг друга элементов. Можно реализовать все эти конъюнкции совместно.

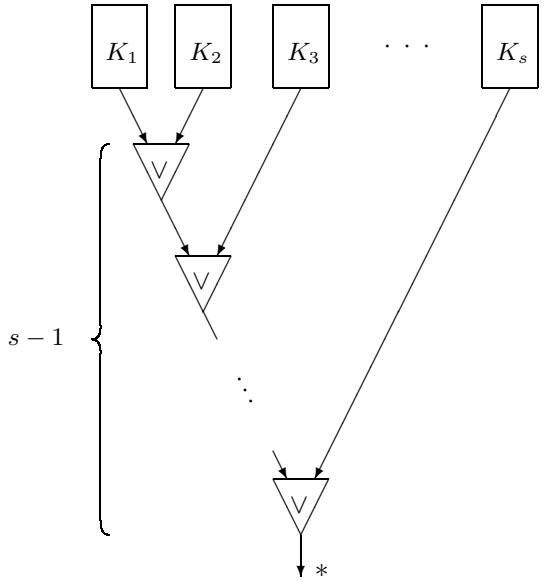


Рис. 8

Пусть $\mathcal{K}_n(x_1,\ldots,x_n)$ — система всех 2^n конъюнкций $x_1^{\sigma_1}\ldots x_n^{\sigma_n}$. Обозначим через $L(\mathcal{K}_n)$ сложность реализации этой системы функций схемами из функциональных элементов. Из леммы 2 следует, что

$$L(\mathcal{K}_n) \leqslant n2^{n+1}.$$

Установим более сильную оценку

Лемма 3. *Имеет место соотношение*[2)]

$$L(\mathcal{K}_n) \sim 2^n.$$

[2)] Соотношение $a(n) \lesssim b(n)$ означает, что $\overline{\lim}_{n\to\infty} \frac{a(n)}{b(n)} \leqslant 1$, а соотношение $a(n) \sim b(n)$ — что $\lim_{n\to\infty} \frac{a(n)}{b(n)} = 1$.

Доказательство. Каждая конъюнкция $x_1^{\sigma_1}\ldots x_n^{\sigma_n}$ может быть представлена в виде конъюнкции двух конъюнкций длины k и $n-k$:

$$x_1^{\sigma_1}\ldots x_n^{\sigma_n} = (x_1^{\sigma_1}\ldots x_k^{\sigma_k})(x_{k+1}^{\sigma_{k+1}}\ldots x_n^{\sigma_n}).$$

Поэтому схема для $\mathcal{K}_n$ может быть образована из схем для $\mathcal{K}_k(x_1,\ldots,x_k)$ и $\mathcal{K}_{n-k}(x_{k+1},\ldots,x_n)$ и системы из 2^n элементов конъюнкции, осуществляющих вышеприведенную операцию (рис. 9). Следовательно,

$$L(\mathcal{K}_n) \leqslant L(\mathcal{K}_k) + L(\mathcal{K}_{n-k}) + 2^n.$$

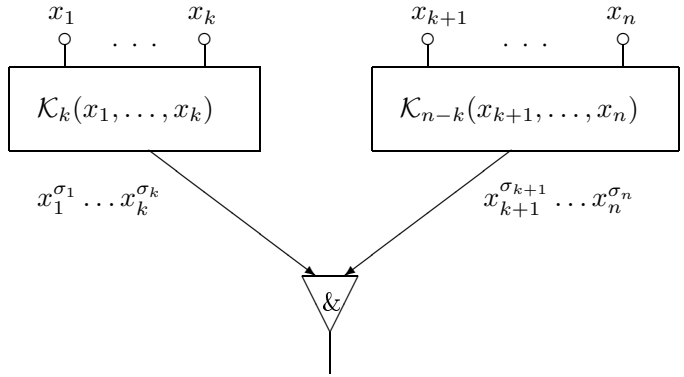


Рис. 9

Так как $L(\mathcal{K}_k) \leqslant k2^{k+1}$, $L(\mathcal{K}_{n-k}) \leqslant (n-k)2^{n-k+1}$, то

$$L(\mathcal{K}_n) \leqslant k2^{k+1} + (n-k)2^{n-k+1} + 2^n.$$

Положим $k = [\frac{n}{2}]$. Тогда[3)] $k \leqslant \frac{n}{2}$, $n-k \leqslant \frac{n}{2}+1$ и

$$L(\mathcal{K}_n) \leqslant \frac{n}{2}2^{\frac{n}{2}+1} + (\frac{n}{2}+1)2^{\frac{n}{2}+2} + 2^n = 2^n + O(n2^{\frac{n}{2}}).$$

С другой стороны, при $n \geqslant 2$ каждая конъюнкция реализуется на выходе некоторого элемента, т. е. при $n \geqslant 2$ выполняется неравенство $L(\mathcal{K}_n) \geqslant 2^n$. Таким образом,

$$L(\mathcal{K}_n) \sim 2^n.$$

[3)] Соотношение $a(n) = O(b(n))$ означает, что существует положительная константа c, такая, что при достаточно больших n выполняется неравенство $a(n) \leqslant cb(n)$.

Теорема 2. *Имеет место соотношение*

$$L(n) \lesssim 2^{n+1}.$$

Доказательство. Пусть $f(x_1, \ldots, x_n)$ — произвольная булева функция, $f \neq 0$. Заменим в схеме рис. 8 верхнюю часть схемы, реализующую конъюнкции $K_1, \ldots, K_s$, схемой, реализующей все конъюнкции из $\mathcal{K}_n$. Тогда для любой такой функции $f(x_1, \ldots, x_n)$ (не равной нулю) имеем

$$L(f) \leqslant L(\mathcal{K}_n) + s - 1 \leqslant L(\mathcal{K}_n) + 2^n - 1 \lesssim 2^{n+1}.$$

Таким образом,

$$L(n) \lesssim 2^{n+1}.$$

Приведем теперь метод, предложенный К. Э. Шенноном[4)] в 1949 г. для контактных схем.

Теорема 3. *Имеет место соотношение*

$$L(n) \lesssim 12 \cdot \frac{2^n}{n}.$$

Доказательство. Пусть $f(x_1, \ldots, x_n)$ — произвольная булева функция. Рассмотрим разложение f по переменным $x_1, \ldots, x_m$, где $1 \leqslant m \leqslant n$:

$$f(x_1, \ldots, x_n) = \bigvee_{(\sigma_1, \ldots, \sigma_m)} x_1^{\sigma_1} \ldots x_m^{\sigma_m} f(\sigma_1, \ldots, \sigma_m, x_{m+1}, \ldots, x_n).$$

Схема для функции f строится из трех подсхем: S_1, S_2 и S_3 (рис. 10). Схема S_1 реализует все конъюнкции из множества $\mathcal{K}_m(x_1, \ldots, x_m)$. В силу леммы 3 выполняется неравенство

$$L(S_1) \leqslant L(\mathcal{K}_m) \lesssim 2^m.$$

Схема S_2 реализует систему $F(x_{m+1}, \ldots, x_n)$ всех булевых функций от переменных $x_{m+1}, \ldots, x_n$. В силу теоремы 1

$$L(S_2) \leqslant (n-m)2^{n-m+1}2^{2^{n-m}}.$$

[4)] *Shannon C. E.* The synthesis of two-terminal switching circuits// Bell Syst. Techn. J. 1949. **28**, N 1. 59-98 (рус. пер.: *Шеннон К.* Работы по теории информации и кибернетики. М.: ИЛ, 1963. 59–101).

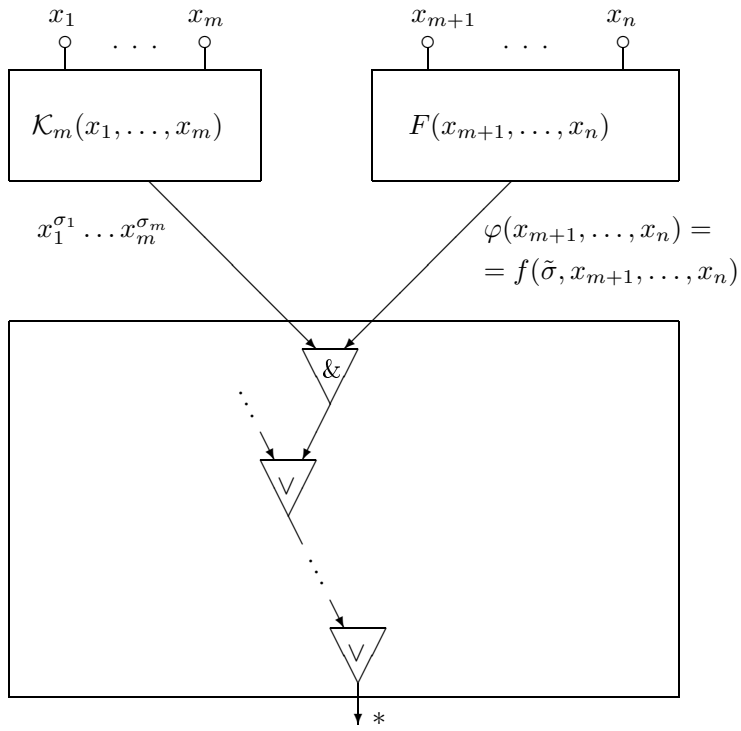


Рис. 10

Схема S_3 производит "сборку" в соответствии с разложением функции f: для каждого набора $\tilde{\sigma} = (\sigma_1, \ldots, \sigma_m)$ реализуется конъюнкция

$$x_1^{\sigma_1} \ldots x_m^{\sigma_m} f(\tilde{\sigma}, x_{m+1}, \ldots, x_n)$$

(2^m элементов конъюнкции) и образуется дизъюнкция таких конъюнкций ($2^m - 1$ элементов дизъюнкции). Поэтому выполняется неравенство $L(S_3) \leqslant 2^m + 2^m - 1$. Таким образом,

$$L(S) = L(S_1) + L(S_2) + L(S_3) \lesssim 3 \cdot 2^m + (n-m)2^{n-m+1}2^{2^{n-m}}.$$

Положим (для упрощения дальнейших выкладок) $k = n - m$. Тогда

$$L(n) \lesssim 3 \cdot 2^{n-k} + k2^{k+1}2^{2^k}.$$

Для минимизации этого выражения можно было бы попытаться применить методы математического анализа (взять производную, найти ее нуль и т. д.). Однако, во-первых, получилось бы сложное уравнение и, во-вторых, мы уже огрубляли оценку, заменив величину $L(\mathcal{K}_m)$ ее асимптотическим выражением. Поэтому будем искать "приблизительный минимум". Заметим, что второе слагаемое "очень быстро" растет с ростом k (двойная экспонента от k), а первое слагаемое убывает с ростом k медленней. Поэтому, по-видимому, следует взять такое значение k, при котором первое и второе слагаемые "приблизительно" равны, и потом "немного" уменьшить k. Тогда второе слагаемое "сильно" уменьшится, а первое "не очень сильно" возрастет. Возьмем, например, $k = \log_2 n$. Тогда

$$3 \cdot 2^{n-k} = 3 \cdot \frac{2^n}{n}, \quad k \cdot 2^{k+1} \cdot 2^{2^k} = \log_2 n \cdot (2n) \cdot 2^n,$$

т. е. получили "слишком много". Возьмем k на единицу меньше: $k = \log_2 n - 1$. Тогда

$$3 \cdot 2^{n-k} = 3 \cdot \frac{2^n}{n} \cdot 2, \quad k \cdot 2^{k+1} \cdot 2^{2^k} = (\log_2 n - 1) \cdot n \cdot 2^{\frac{n}{2}}.$$

Вспомним теперь, что k должно быть целым числом, и положим $k = [\log_2 n - 1]$. Тогда

$$n - k < n - \log_2 n + 2, \quad 3 \cdot 2^{n-k} < 12 \cdot \frac{2^n}{n},$$

$$k \cdot 2^{k+1} \cdot 2^{2^k} \leqslant (\log_2 n - 1) \cdot n \cdot 2^{\frac{n}{2}}.$$

При этом выборе k окончательно имеем

$$L(n) \lesssim 12 \cdot \frac{2^n}{n}.$$

Тем самым теорема доказана.

Заметим, что можно было бы улучшить эту оценку, уточнив константу (при $k = [\log_2(n - 3\log_2 n)]$ получим $L(n) \lesssim 6 \cdot \frac{2^n}{n}$).

Лекция № 8

Покажем, что порядок верхней оценки функции $L(n)$, полученной методом Шеннона, не может быть уменьшен. При этом не будет указан пример конкретной функции $f(x_1, \ldots, x_n)$, допускающей лишь сложную реализацию, но будет показано, что количество "просто реализуемых" функций $f(x_1, \ldots, x_n)$ меньше, чем число всех функций от n аргументов. Это доказательство является "конечным аналогом" доказательства существования трансцендентных чисел: алгебраических чисел (корней алгебраических уравнений с целыми коэффициентами) — счетное множество, всех действительных чисел — континуум; поэтому существуют действительные числа, не являющиеся алгебраическими, т. е. трансцендентные числа. Такое доказательство существования сложно реализуемых функций алгебры логики впервые было использовано в работе Дж. Риордана и К. Шеннона [1].

Пусть $N(n, h)$ — число функций $f(x_1, \ldots, x_n)$, допускающих реализацию схемами сложности не более h, а $N'(n, h)$ — число функций $f(x_1, \ldots, x_n)$, допускающих реализацию схемами сложности h. Очевидно, что $N(n, h) = N'(n, h)$, так как если функция f допускает реализацию схемой сложности h', $h' \leqslant h$, то она допускает реализацию и схемой сложности в точности h: для этого к схеме S достаточно присоединить $h - h'$ элементов, выходы которых "ни к чему" не присоединяются.

Обозначим через $N''(n, h)$ число схем сложности h с одним выходом, реализующих функции алгебры логики от переменных $x_1, \ldots, x_n$. Очевидно, что $N'(n, h) \leqslant N''(n, h)$.

Лемма. *Имеет место неравенство*

$$N''(n, h) \leqslant 3^h (n + h)^{2h+1}.$$

Доказательство. Возьмем n вершин, помеченных символами $x_1, \ldots, x_n$, — входы схемы и h непомеченных вершин — элементы схемы. Каждый из элементов пометим одним из символов $\&, \vee, {}^-$; имеем 3^h возможностей. Тем самым определится число входов всех элементов; это число не превосходит $2h$. Каждый вход присоединим либо к одному из входов схемы, либо к одному из выходов

[1] См.: *Riordan J., Shannon C. E.* The number of two-terminal series-parallel networks // J. Math. and Phys. 1942. **21**, N 2. 83–93 (рус. пер.: *Шеннон К.* Работы по теории информации и кибернетики. М.: ИЛ, 1963. 46–58).

элементов — $n+h$ возможностей для каждого входа; число способов присоединения для всех входов не превосходит $(n+h)^{2h}$. Кроме того, одна из вершин может быть помечена символом $*$ — $n+h$ возможностей. Среди получившихся конфигураций окажется большое число бессмысленных (содержащих ориентированные циклы), но все "настоящие" схемы среди них будут. Таким образом,

$$N''(n,h) \leqslant 3^h (n+h)^{2h+1}.$$

Теорема 1. *При достаточно больших n выполняется неравенство*

$$L(n) > \frac{1}{3} \cdot \frac{2^n}{n}.$$

Доказательство. Очевидно, что если $N(n,h_0) < 2^{2^n}$ для некоторого h_0, то $L(n) > h_0$. Покажем, что при $h_0 = \frac{1}{3} \cdot \frac{2^n}{n}$ выполняется неравенство $N''(x,h_0) < 2^{2^n}$. В самом деле, используя лемму 1, имеем[2)]

$$\log_2 \tfrac{N''(n,h_0)}{2^{2^n}} = \log_2 N''(n,h_0) - 2^n \leqslant$$

$$\leqslant (2h_0+1)\log_2(n+h_0) + h_0 \log_2 3 - 2^n \leqslant$$

$$\leqslant (\tfrac{2}{3}\cdot\tfrac{2^n}{n}+1)\log_2(n+\tfrac{1}{3}\cdot\tfrac{2^n}{n}) + \tfrac{\log_2 3}{3}\cdot\tfrac{2^n}{n} - 2^n <\cdot$$

$$<\cdot (\tfrac{2}{3}\cdot\tfrac{2^n}{n}+1)\log_2(2^n) + \tfrac{\log_2 3}{3}\cdot\tfrac{2^n}{n} - 2^n =$$

$$= \tfrac{\log_2 3}{3}\cdot\tfrac{2^n}{n} + n - \tfrac{1}{3}\cdot 2^n \to -\infty$$

при $n \to \infty$. Поэтому

$$\frac{N''(n,h_0)}{2^{2^n}} \to 0$$

(при $n \to \infty$) и $N''(n,h_0) <\cdot 2^{2^n}$. Отсюда и из неравенства $N(n,h_0) \leqslant N''(n,h_0)$ получаем утверждение теоремы.

Таким образом, функция $L(n)$ по порядку равна $\frac{2^n}{n}$. На самом деле можно доказать, что[3)]

$$L(n) \sim \frac{2^n}{n}.$$

[2)] Соотношение $a(n) <\cdot b(n)$ означает, что неравенство справедливо при достаточно больших n.

[3)] См.: *Лупанов О. Б.* О синтезе некоторых классов управляющих систем // Проблемы кибернетики. Вып. 10. М.: Физматгиз, 1963. 63-97.

Контактные схемы. Рассмотрим конечный неориентированный граф G и конечный алфавит $X = \{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n\}$. *Контактной схемой* называется неориентированный граф с выделенными вершинами — *полюсами*, каждому ребру которого приписана некоторая буква алфавита X (разным ребрам могут соответствовать одинаковые буквы). Ребро вместе с приписанной ему буквой x^σ ($\sigma = 0, 1$) называется *контактом*, причем *замыкающим* контактом, если $\sigma = 1$, и *размыкающим* контактом, если $\sigma = 0$.

Каждой паре полюсов (a,b) сопоставим *функцию проводимости* $f_{ab}(x_1, \ldots, x_n)$ (между полюсами a и b) следующим образом:

1) если $a = b$, то $f_{ab}(x_1, \ldots, x_n) = 1$;

2) если $a \neq b$ и в схеме не существует цепей между a и b, то $f_{ab}(x_1, \ldots, x_n) = 0$;

3) если $a \neq b$ и множество цепей между a и b непусто, то сопоставим каждой цепи I вида

$$(a, v_{i_1}),\ (v_{i_1} v_{i_2}),\ \ldots,\ (v_{i_{m-1}}, b)$$

конъюнкцию $x_{r_1}^{\sigma_1} \ldots x_{r_m}^{\sigma_m}$ (*проводимость* цепи I), где $x_{r_t}^{\sigma_t}$ — буква из X, приписанная ребру $(v_{i_{t-1}}, v_{i_t})$ (здесь $v_{i_0} = a$, $v_{i_m} = b$). Положим

$$f_{ab}(x_1, \ldots, x_n) = \bigvee_I x_{r_1}^{\sigma_1} \ldots x_{r_m}^{\sigma_m},$$

где дизъюнкция берется по всем цепям I между a и b.

Будем говорить также, что схема реализует сопоставленные ей функции.

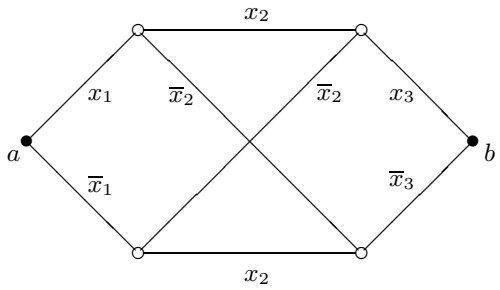


Рис. 1

Пример. Схема (с полюсами a и b), изображенная на рис. 1, реализует функцию

$$f_{ab}(x_1,x_2,x_3) = x_1x_2x_3 \vee \overline{x}_1x_2\overline{x}_3 \vee x_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1\overline{x}_2x_3 =$$

$$= x_1 \oplus x_2 \oplus x_3.$$

Так как граф неориентированный, то $f_{ab} = f_{ba}$.

Для каждой контактной схемы можно составить матрицу функций проводимости, в которую поместим функции проводимости для соответствующих пар полюсов. Для схемы, изображенной на рис. 1, эта матрица выглядит следующим образом:

	a	b
a	1	$x_1 \oplus x_2 \oplus x_3$
b	$x_1 \oplus x_2 \oplus x_3$	1

Пусть в схеме имеется k полюсов $a_1, a_2, \ldots, a_k$. Возникает вопрос: какие условия необходимо наложить на произвольную квадратную матрицу M порядка k, в клетках которой расположены булевы функции, чтобы существовала такая контактная схема, у которой матрица функций проводимости совпадает с M? Прежде всего заметим, что для любых трех полюсов a_i, a_j и a_k функция $f_{a_ia_k}$ должна принимать значение 1 на всех наборах значений переменных, на которых функции $f_{a_ia_j}$ и $f_{a_ja_k}$ одновременно принимают значение 1. Это означает, что $f_{a_ia_k} \geqslant f_{a_ia_j}\&f_{a_ja_k}$. Очевидно, что матрица должна быть симметрична относительно главной диагонали (так как графы неориентированные). Кроме того, на главной диагонали должны стоять единицы. Оказывается, что этих трех необходимых условий достаточно для существования контактной схемы с заданной матрицей функции проводимости.

Сложность контактной схемы S определим как число ее контактов; обозначим эту величину через $L(S)$. Определим далее функции, аналогичные введенным ранее для схем из функциональных элементов:

$$L_k(f) = \min L(S), \quad L_k(n) = \max L_k(f),$$

где минимум берется по всем контактным схемам, реализующим функцию f, а максимум — по всем функциям f от аргументов $x_1, \ldots, x_n$.

Очевидно, что если схема S_1 реализует функцию f_1, а схема S_2 — функцию f_2, то в результате параллельного (соответственно последовательного) соединения этих схем получается схема, реализующая $f_1 \vee f_2$ (соответственно $f_1\&f_2$).

Покажем, что любая функция $f(x_1, \ldots, x_n)$ может быть реализована контактной схемой. Рассмотрим совершенную дизъюнктивную нормальную форму этой функции. Пусть $\tilde{\sigma}^1 = (\sigma_1^1, \ldots, \sigma_n^1), \ldots, \tilde{\sigma}^s = (\sigma_1^s, \ldots, \sigma_n^s)$ — все наборы, на которых функция f принимает значение 1, $s \leqslant 2^n$. Для каждого набора $\tilde{\sigma}^i$ построим цепь, соединяющую два полюса схемы (a и b) и реализующую конъюнкцию $x_1^{\sigma_1^i} \ldots x_n^{\sigma_n^i}$, и соединим все эти цепи параллельно. В результате получим схему S, которая реализует функцию f (рис. 2). Очевидно, что $L(S) \leqslant n2^n$.

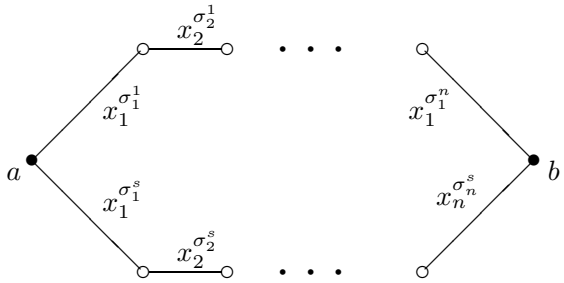


Рис. 2

Для функции, тождественно равной нулю, можно взять схему, состоящую из двух изолированных полюсов. Таким образом, имеет место

Теорема 2. *Справедливо неравенство*

$$L_k(n) \leqslant n2^n.$$

Приведенное математическое определение функции, реализуемой контактной схемой, отражает "физическую" сторону дела. Например, схема на рис. 1 соответствует "физической схеме", изображенной на рис. 3.

Контакт на рис. 4, *а* разомкнут при невозбужденной обмотке реле и замкнут при возбужденной обмотке; контакт, изображенный на рис. 4, *б*, наоборот, замкнут при невозбужденной обмотки реле и разомкнут при возбужденной. Считаем, что обмотка возбуждена тогда и только тогда, когда соответствующая переменная равна 1.

Тогда контакт "физической схемы", соответствующий контакту x^σ (математической модели), разомкнут, когда $x^\sigma = 0$, и замкнут, когда $x^\sigma = 1$. Поэтому для каждого набора значений переменных значение функции, реализуемой контактной схемой, равно 1 тогда и только тогда, когда "физическая схема" проводит ток.

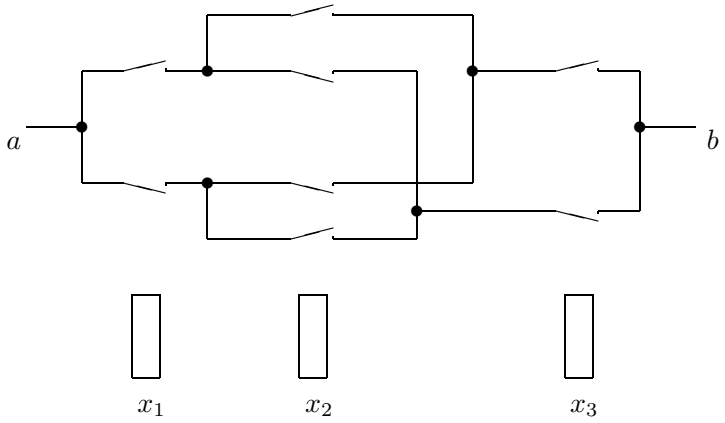


Рис. 3

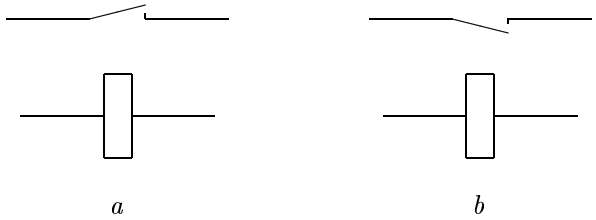


Рис. 4

Контактная схема называется $(1,k)$-*полюсником*, если в ней некоторый полюс считается *входным*, а остальные k полюсов — *выходными* (при этом допускается, что полюс может быть входным и выходным одновременно). С каждым $(1,k)$-полюсником связывается система из k функций, каждая из которых реализуется между входом и некоторым выходом. Если проводимость между любыми

двумя различными выходами $(1,k)$-полюсника тождественно равна нулю, то он называется *разделительным*. *Контактным деревом* называется такой $(1,2^n)$-полюсник, который реализует (между входом и выходами) систему $\mathcal{K}(x_1,\ldots,x_n)$ всех 2^n конъюнкций вида $x_1^{\sigma_1}\ldots x_n^{\sigma_n}$. Пример контактного дерева изображен на рис. 5.

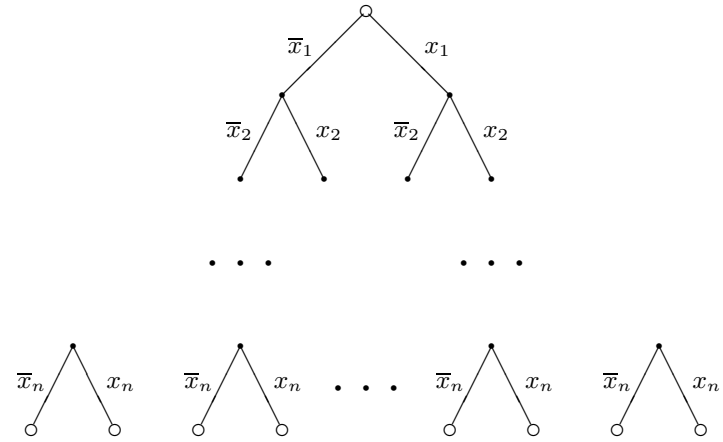


Рис. 5

Легко видеть, что данное контактное дерево содержит

$$2+4+\cdots+2^n = 2^{n+1}-2$$

контактов.

Заметим, что контактное дерево, приведенное на рис. 5, является разделительным $(1,2^n)$-полюсником. Можно доказать, что в классе разделительных контактных схем данное контактное дерево является минимальным. Вместе с тем если отказаться от свойства разделительности, то систему всех конъюнкций $\mathcal{K}(x_1,\ldots,x_n)$ можно реализовать со сложностью, асимптотически равной 2^n.

На основе контактного дерева можно предложить следующий метод синтеза контактных схем. Пусть $f(x_1,\ldots,x_n)$ — произвольная булева функция. Если $f \neq 0$, то f может быть задана совершенной дизъюнктивной нормальной формой

$$f(x_1,\ldots,x_n) = K_1 \vee K_2 \vee \cdots \vee K_s,$$

где $s \leqslant 2^n$ и каждая конъюнкция имеет вид $K_j = x_1^{\sigma_{j_1}} x_2^{\sigma_{j_2}} \dots x_n^{\sigma_{j_n}}$. Возьмем построенное выше контактное дерево (рис. 5), реализующее все конъюнкции вида $x_1^{\sigma_1} \dots x_n^{\sigma_n}$. Отождествим в нем все выходы, соответствующие конъюнкциям $K_1, K_2, \dots, K_s$. Получим контактную схему S. Поскольку проводимость между любыми двумя различными выходами контактного дерева равна нулю, то функция проводимости схемы S между входом и новым выходом равна f. При этом $L(S) \leqslant 2^{n+1}-2$, т. е. $L_k(f) \leqslant 2^{n+1}$. Тем самым доказана

Теорема 3. *Имеет место неравенство*

$$L_k(n) \leqslant 2^{n+1}.$$

Приведем теперь метод построения контактных схем и схем из функциональных элементов, учитывающий специфику конкретных функций. Этот метод получил название "метод каскадов" и был описан в упомянутой ранее работе К. Шеннона в 1949 г. Метод каскадов учитывает некоторые свойства функций и благодаря этому позволяет для многих функций строить сравнительно простые схемы. Метод состоит из двух этапов. Сначала исходная функция разлагается по переменным и строится последовательность множеств функций, а затем на основе этой последовательности строится соответствующая схема.

Пусть дана функция $f(x_1, \dots, x_n)$. Образуем последовательность множеств $G_0, G_1, \dots, G_{n-1}$, где

$$G_i = \{g_{i,1}(x_{i+1}, \dots, x_n), \dots, g_{i,r_i}(x_{i+1}, \dots, x_n)\},$$

для каждого $i = 0, 1, \dots, n-1$ G_i — множество функций от переменных $x_{i+1}, \dots, x_n$. Определим множества G_i индукцией по i. Положим

$$G_0 = \{f(x_1, \dots, x_n)\} = \{g_{0,1}(x_1, \dots, x_n)\}.$$

Пусть множество G_i уже построено. Рассмотрим следующие $2r_i$ функций:

$$g_{i,1}(0, x_{i+2}, \dots, x_n), \dots, g_{i,r_i}(0, x_{i+2}, \dots, x_n),$$
$$g_{i,1}(1, x_{i+2}, \dots, x_n), \dots, g_{i,r_i}(1, x_{i+2}, \dots, x_n).$$

Множество G_{i+1} по определению состоит из всех различных функций, встречающихся среди этих $2r_i$ функций. В частности, G_{n-1} есть некоторое подмножество множества $\{x_n, \overline{x}_n, 0, 1\}$.

Из определения множеств G_i вытекает следующее свойство: каждая функция $g(x_{i+1}, \dots, x_n)$ из G_i, $0 \leqslant i \leqslant n-2$, может быть представлена в виде

$$g(x_{i+1}, \dots, x_n) = x_{i+1} g^{(1)}(x_{i+2}, \dots, x_n) \vee \overline{x}_{i+1} g^{(2)}(x_{i+2}, \dots, x_n),$$

где $g^{(1)}, g^{(2)} \in G_{i+1}$.

Заметим, что $|G_0| = r_0 = 1$ и $|G_{i+1}| = r_{i+1} \leqslant 2r_i$ для всех $i = 0, 1, \dots, n-2$, т. е. число функций в каждом последующем классе не более чем в два раза превышает число функций в предыдущем классе. Поэтому $r_i \leqslant 2^i$, $i = 0, 1, \dots, n-1$. С другой стороны, $r_i \leqslant 2^{2^{n-i}}$, поскольку G_i — подмножество множества всех булевых функций от переменных $x_{i+1}, \dots, x_n$.

Схема строится в соответствии с описанным выше разложением, но в "обратном порядке". А именно строится последовательность контактных схем $S_{n-1}, S_{n-2}, \dots, S_0$, где S_i — $(1, r_i)$-полюсник, между входом и r_i выходами которого реализуются все функции из множества G_i, $i = 0, 1, \dots, n-1$.

Так как $G_{n-1} \subseteq \{x_n, \overline{x}_n, 0, 1\}$, то в качестве схемы S_{n-1} можно взять схему, приведенную на рис. 6 (выход, на котором реализуется 1, совпадает с входом).

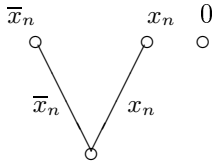


Рис. 6

Пусть уже построена схема S_{i+1} ($(1, r_{i+1})$-полюсник). Построим схему S_i следующим образом. В качестве ее входа возьмем вершину a — вход схемы S_{i+1}, а в качестве выходов возьмем r_i новых вершин $v_1, \dots, v_{r_i}$; вершинам $v_1, \dots, v_{r_i}$ соответствуют функции $g_{i,1}, \dots, g_{i,r_i}$ из множества G_i. Пусть $g_{ij} \in G_i$, $j = 1, \dots, r_i$. Из приведенного выше свойства множеств G_i следует, что найдутся функции $g^{(1)}$ и $g^{(2)}$ из G_{i+1}, такие, что

$$g_{ij}(x_{i+1}, \dots, x_n) = x_{i+1} g^{(1)}(x_{i+2}, \dots, x_n) \vee \overline{x}_{i+1} g^{(2)}(x_{i+2}, \dots, x_n).$$

Поэтому в схеме S_{i+1} найдутся выходы $w^{(1)}$ и $w^{(2)}$, на которых реализуются функции $g^{(1)}$ и $g^{(2)}$ соответственно (т. е. $f_{aw^{(1)}} = g^{(1)}$,

$f_{aw^{(2)}} = g^{(2)}$). Соединим вершины v_j и $w^{(1)}$ контактом x_{i+1}, а вершины v_j и $w^{(2)}$ — контактом $\overline{x}_{i+1}$. Очевидно, что функция проводимости f_{av_j} между полюсами a и v_j равна g_{ij}.

Итак, мы построили $(1, r_i)$-полюсник — схему S_i. Легко видеть, что поскольку из каждого выхода $v_1, \ldots, v_{r_i}$ схемы S_i выходят ровно два контакта x_{i+1} и $\overline{x}_{i+1}$, то между вершиной a и выходами $v_1, \ldots, v_{r_i}$ будут реализованы все функции из множества G_i (см. рис. 7).

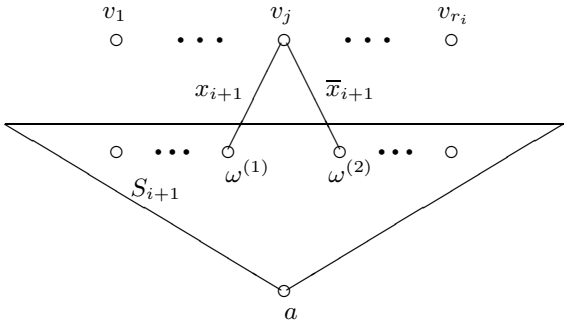


Рис. 7

Таким образом, схема S_0 реализует функцию f. Поэтому $L_k(f) \leqslant L(S_0)$. Так как $L(S_{n-1}) = 2$ и $L(S_{i+1}) + 2r_i$, $i = 0, 1, \ldots, n-2$, то

$$L_k(f) \leqslant 2 + 2\sum_{i=0}^{n-2} r_i.$$

Заметим, что для любого $k = 1, \ldots, n-1$ выполняются неравенства

$$L(S_0) \leqslant 2 \sum_{i=0}^{n-k-1} r_i + L(S_{n-k}) \leqslant 2 \cdot 2^{n-k} + L(S_{n-k}),$$

где S_{n-k} — схема, реализующая все функции $g(x_{n-k+1}, \ldots, x_n)$ из множества G_{n-k}. Заменим в схеме S_0 подсхему S_{n-k} на $(1, 2^{2^k})$-полюсник U_k, реализующий все функции от k переменных $x_{n-k+1}, \ldots, x_n$. В силу теоремы 3 $L(U_k) \leqslant 2^{k+1}2^{2^k}$. В результате получим схему S, реализующую функцию f; при этом

$$L(S) \leqslant 2 \cdot 2^{n-k} + 2^{k+1} \cdot 2^{2^k}.$$

Поэтому выполняется неравенство $L_k(f) \leqslant 2 \cdot 2^{n-k} + 2^{k+1} \cdot 2^{2^k}$. Положим $k = [\log_2 n - 1]$. Тогда

$$L_k(n) \lesssim 8 \cdot \frac{2^n}{n}.$$

Так же и для схем из функциональных элементов нетрудно доказать, что полученная верхняя оценка функции $L_k(n)$ является наилучшей по порядку.

Пусть $N_k(n, h)$ — число булевых функций $f(x_1, \ldots, x_n)$, допускающих реализацию контактными схемами сложности не более h, а $N'_k(n, h)$ — число функций $f(x_1, \ldots, x_n)$, допускающих реализацию контактными схемами сложности h. Очевидно, что $N_k(n, h) = N'_k(n, h)$.

Обозначим через $N''_k(n, h)$ число двухполюсных контактных схем с h ребрами, которые не содержат изолированных вершин, отличных от полюсов, и у которых ребра помечены символами из множества $\{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n\}$. Легко видеть, что выполняется неравенство $N'_k(n, h) \leqslant N''_k(n, h)$.

Получим верхнюю оценку для $N''_k(n, h)$.

Каждая контактная схема с h ребрами, в которой нет изолированных вершин, отличных от полюсов, содержит не более $2h + 2$ вершин. Каждое ребро можно провести не более чем $(2h + 2)\,(2h + 2)$ способами, число способов провести h ребер не превосходит $(2h + 2)^{2h}$. Каждое из ребер можно пометить одним из $2n$ символов $x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n$; имеем $(2n)^h$ возможностей. Кроме того, две вершины выбираются в качестве полюсов; имеем не более $(2h + 2)^2$ способов. Таким образом,

$$N''(n, h) \leqslant (2h + 2)^2 \cdot (2n)^h \cdot (2h + 2)^{2h} = (2n)^h \cdot (2h + 2)^{2h+2}.$$

Далее легко показать, что при $h_0 = \frac{1}{3} \cdot \frac{2^n}{n}$ для достаточно больших n выполняется неравенство

$$N''(n, h_0) < 2^{2^n}.$$

Поэтому

$$L_k(n) \gtrsim \frac{1}{3} \cdot \frac{2^n}{n}.$$

Таким образом, имеет место

Теорема 4. *Справедливо соотношение*[4)]

$$L_k(n) \asymp \frac{2^n}{n}.$$

Итак, функций $L_k(n)$ по порядку равна $\frac{2^n}{n}$. На самом деле можно доказать, что[5)]

$$L_k(n) \sim \frac{2^n}{n}.$$

Пример. Для функции $x_1 \oplus \cdots \oplus x_n$, $n \geqslant 2$, множество G_0 состоит из одной функции $x_1 \oplus \cdots \oplus x_n$, а множество G_i при $1 \leqslant i \leqslant n-1$ — из двух функций $x_{i+1} \oplus \cdots \oplus x_n$ и $x_{i+1} \oplus \cdots \oplus x_n \oplus 1$. Соответствующая схема изображена на рис. 8.

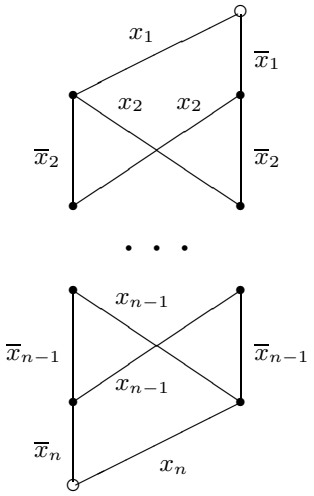


Рис. 8

Сложность данной схемы равна $2 \cdot 2 + 4(n-2) = 4n-4$. Можно доказать, что она является минимальной. Этот результат получил К. Кардо в 1952 г., хотя сама схема была известна еще Шеннону, который привел ее в своей работе 1949 г.

[4)] Соотношение $a(n) \asymp b(n)$ означает, что выполняются соотношения $a(n) \preccurlyeq b(n)$ и $b(n) \preccurlyeq a(n)$; соотношение $a(n) \preccurlyeq b(n)$ означает, что существует положительная константа c, такая, что $a(n) <\cdot\, cb(n)$ (т. е. неравенство справедливо при достаточно больших n).

[5)] См.: *Лупанов О. Б.* О синтезе некоторых классов управляющих систем// Проблемы кибернетики. Вып. 10. М.: Физматгиз, 1963. 63-97.

Метод каскадов можно применить и для построения схем из функциональных элементов, реализующих заданную функцию f. А именно будем последовательно строить схемы из функциональных элементов $S_{n-1}, S_{n-2}, \ldots, S_0$, реализующих функции из множеств $G_{n-1}, G_{n-2}, \ldots, G_0$ соответственно. В качестве схемы S_{n-1} можно взять, например, схему, изображенную на рис. 9.

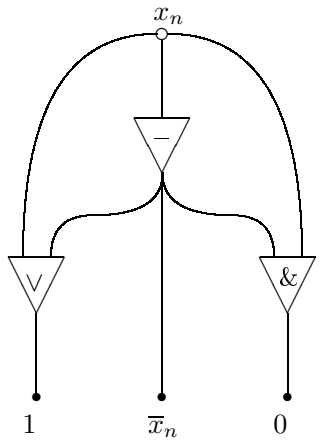


Рис. 9

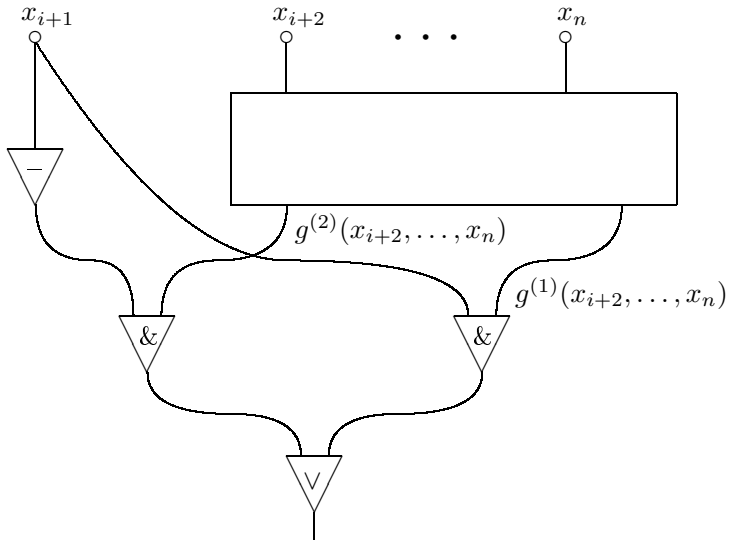
При $i = 0, 1, \ldots, n-2$ схема S_i строится на основе схемы S_{i+1} с использованием одного дополнительного элемента отрицания для всего множества G_i и одного элемента дизъюнкции и двух элементов конъюнкции для каждой функцию $g(x_{i+1}, \ldots, x_n)$ из множества G_i (см. рис. 10).

Таким образом, $L(S_{n-1}) = 3$ и $L(S_i) = L(S_{i+1}) + 1 + 3r_i$ при $i = 0, 1, \ldots, n-2$. Поэтому

$$L(S_0) = 3 + 3\sum_{i=0}^{n-2} r_i + n - 1 = n + 2 + 3\sum_{i=0}^{n-2} r_i.$$

Так как схема S_0 реализует функцию f, то

$$L(f) \leqslant n + 2 + 3\sum_{i=0}^{n-2} r_i.$$



$$g = x_{i+1}g^{(1)}(x_{i+2},\dots,x_n) \vee \overline{x}_{i+1}g^{(2)}(x_{i+2},\dots,x_n)$$

Рис. 10

Пример. Как мы показали выше, для функции

$$f(x_1,\dots,x_n) = x_1 \oplus \dots \oplus x_n,$$

$n \geqslant 2$, выполняется равенство $r_i = 2$ при всех $i = 1,\dots,n-1$. Поэтому сложность схемы из функциональных элементов, построенной по методу каскадов, для этой функции равна

$$n + 2 + 3\sum_{i=1}^{n-2} 2 + 3 = 7n - 7.$$

Можно показать, что для функции f существует схема из функциональных элементов, сложность которой не превосходит $4n - 4$. Тем самым метод каскадов не приводит к построению минимальной схемы из функциональных элементов для линейной функции.

КОНЕЧНЫЕ АВТОМАТЫ

Лекция № 9

Рассмотрим два конечных алфавита $A = \{a_1,\dots,a_\nu\}$ и $B = \{a_1,\dots,a_\mu\}$. Обозначим через A^∞ и B^∞ множества всех бесконечных последовательностей $\alpha = (\alpha(1),\alpha(2),\dots)$ и $\beta = (\beta(1),\beta(2),\dots)$ в алфавитах A и B соответственно. Рассмотрим функции $f(x)$, определенные на множестве бесконечных последовательностей α из A^∞ и принимающие на каждой из этих последовательностей значение β из B^∞; α и β называются входной и выходной последовательностями функции $f(x)$ соответственно. Таким образом, каждая функция $f(x)$ задает отображение $f : A^\infty \to B^\infty$.

Отметим, что среди этих функций есть такие, при вычислении значений которых возникают определенные трудности. Рассмотрим, например, следующую функцию.

Пусть $A = B = \{0,1\}$. Обозначим через $\tilde{0}$ и $\tilde{1}$ бесконечные последовательности, состоящие из одних нулей и единиц соответственно. Пусть $f(\tilde{0}) = \tilde{0}$, $f(\alpha) = \tilde{1}$ для всех $\alpha \in A^\infty$, $\alpha \neq \tilde{0}$. Тогда значение первого разряда выходной последовательности функции f на последовательности $\tilde{0}$ нельзя найти, зная лишь конечное (каким бы большим оно ни было) число разрядов входной последовательности.

В связи с этим вводим следующее ограничение на рассматриваемые функции. Пусть $y = f(x)$, где

$$x = (x(1),x(2),\dots,x(t),\dots), \quad y = (y(1),y(2),\dots,y(t),\dots).$$

Функция f называется *детерминированной*, если для любого t, $t = 1,2,\dots$, значение $y(t)$ однозначно определяется первыми t членами входной последовательности $x(1),x(2),\dots,x(t)$.

Иными словами, каждая детерминированная функция однозначно определяется бесконечной последовательностью функций

$$f_1(x(1)),\ f_2(x(1),x(2)),\dots,\ f_t((1),x(2),\dots,x(t)),\dots,$$

где $f_t : A^t \to B,\ t = 1,2,\dots$.

Пример. Пусть $A = B = \{0,1\}$. Следующие функции $y = f(x)$, где $f : A^\infty \to B^\infty$, $x = (x(1),\dots,x(t),\dots)$, $y = (y(1),\dots,y(t),\dots)$, являются детерминированными:

a) $y(t) = \begin{cases} 0, \text{ если } x(1) = x(2) = \cdots = x(t-1) = 0; \\ 1 \text{ в противном случае;} \end{cases}$

b) функция четности: $y(t) = x(1) \oplus x(2) \oplus \cdots \oplus x(t)$;

c) функция единичной задержки:
$y(t) = \begin{cases} 0, \text{ если } t = 0; \\ x(t-1) \text{ в противном случае;} \end{cases}$

d) $y(t) = \begin{cases} 0\,, \text{ если } t = 2^n \text{ для некоторого натурального } n; \\ 1 \text{ в противном случае.} \end{cases}$

Детерминированные функции можно задавать при помощи информационных деревьев. *Информационное дерево в алфавитах* A и B представляет собой бесконечное ориентированное дерево, удовлетворяющее следующим условиям:

1) существует вершина v_0 — *корень* информационного дерева, в которую не входит ни одно ребро;
2) в каждую вершину, отличную от корневой, входит ровно одно ребро;
3) из каждой вершины дерева выходит $\nu = |A|$ ребер, которым приписаны пары (a_1, b_{i_1}), $(a_2, b_{i_2}), \ldots,$ (a_ν, b_{i_ν}), где $b_{i_1}, \ldots, b_{i_\nu} \in B$.

Таким образом, любая вершина дерева достижима из корневой и каждой выходящей из корня ориентированной цепи в информационном дереве соответствует пара последовательностей $\alpha \in A^\infty$ и $\beta \in B^\infty$, которые составлены из приписанных ребрам этой цепи букв алфавитов A и B соответственно. Поэтому можно считать, что каждое информационное дерево T в алфавитах A и B задает вполне определенную детерминированную функцию $f_T(x)$; $f_T : A^\infty \to B^\infty$. Легко видеть, что верно и обратное. А именно для каждой детерминированной функции можно построить информационное дерево в алфавитах A и B, которое будет задавать функцию f.

На рис. 1 и 2 изображены начальные фрагменты информационных деревьев в алфавитах $A = \{0, 1\}$ и $B = \{0, 1\}$ для первой и второй функций из приведенного выше примера соответственно.

Возьмем в дереве T произвольную вершину v и рассмотрим бесконечное поддерево T_v дерева T с вершиной v в качестве корня, содержащее все вершины дерева T, достижимые из вершины v.

Легко видеть, что T_v также будет являться информационным деревом в алфавитах A и B, задающим некоторую детерминированную функцию $f_{T_v}(x)$; $f_{T_v} : A^\infty \to B^\infty$.

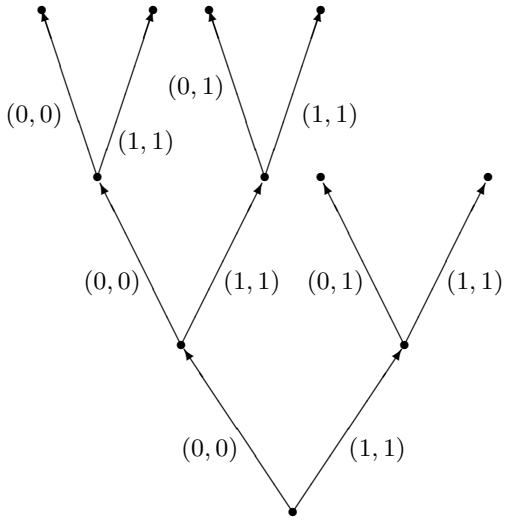


Рис. 1

Два информационных дерева T_1 и T_2 в алфавитах A и B называются *эквивалентными* (обозначение $T_1 \sim T_2$), если они задают одну и ту же детерминированную функцию (т. е. $f_{T_1}(x) = f_{T_2}(x)$). Иными словами, информационные деревья эквивалентны, если существует изоморфизм соответствующих бесконечных деревьев, сохраняющий пометки на ребрах.

Детерминированная функция $f(x)$ называется *ограниченно-детерминированной* (*о.-д. функцией*), если в информационном дереве, задающем функцию $f(x)$, содержится лишь конечное число попарно неэквивалентных информационных поддеревьев. Максимальное число попарно неэквивалентных поддеревьев в информационном дереве, задающем о.-д. функцию $f(x)$, называется *весом* функции f.

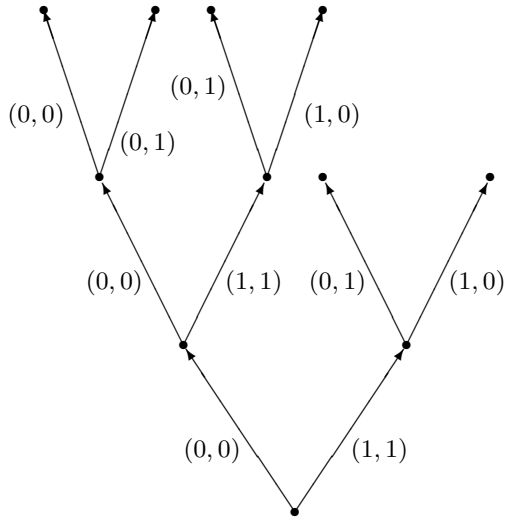


Рис. 2

Пусть $f(x)$ — о.-д. функция веса r, T — информационное дерево, задающее f, v_0 — корень дерева T, а $v_0, v_1, \ldots, v_{r-1}$ — вершины дерева T, такие, что $T_{v_i} \not\sim T_{v_j}$ для всех $i, j = 0, 1, \ldots, r-1$, $i \neq j$. Занумеруем все вершины дерева T числами $0, 1, \ldots, r-1$ следующим образом:

1) вершины $v_0, v_1, \ldots, v_{r-1}$ нумеруются числами $0, 1, \ldots, r-1$ соответственно;
2) каждая вершина $v \notin \{v_0, v_1, \ldots, v_{r-1}\}$ нумеруется числом i, таким, что $T_v \sim T_{v_i}$, $0 \leqslant i \leqslant r-1$.

Номера $0, 1, \ldots, r-1$ вершин $v_0, v_1, \ldots, v_{r-1}$ в информационном дереве T называются *состояниями* функции f; множество $Q = \{0, 1, \ldots, r-1\}$ называется *множеством состояний* о.-д. функции f.

Вес функции четности из приведенного выше примера равен 2. На рис. 3 указан начальный фрагмент задающего эту функцию

информационного дерева, у которого вершины занумерованы числами 0 и 1; для наглядности вершины дерева изображены в виде кружков, внутри которых помещены соответствующие номера.

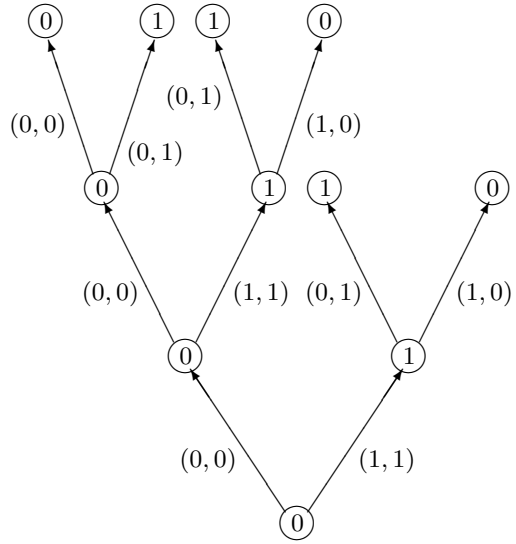


Рис. 3

Полученное информационное дерево T с занумерованными числами $0, 1, \ldots, r-1$ вершинами содержит избыточные сведения об исходной о.-д. функции f; вся необходимая информация содержится в r конечных фрагментах дерева, представленного на рис. 4, где $b_{i_1}, \ldots, b_{i_v} \in B$, а $j_1, j_2, \ldots, j_v \in Q$, $i = 0, 1, \ldots, r-1$.

На рис. 5 изображены соответствующие фрагменты для информационного дерева с занумерованными вершинами, изображенного на рис. 3.

Эта информация содержится также в усеченном дереве. *Усеченное дерево* представляет собой конечное ориентированное дерево, является подграфом дерева T с сохранением всех пометок на вершинах и ребрах, содержит корень v_0 дерева T и обладает

следующим свойством: любая ориентированная цепь, выходящая из корня, содержит ровно две вершины с одинаковыми номерами, а никакое его собственное поддерево этим свойством не обладает. Легко видеть, что усеченное дерево содержит все упомянутые выше фрагменты дерева T (см. рис. 4); при этом число ребер в любой ориентированной цепи этого дерева не превышает r. Таким образом, усеченное дерево содержит всю необходимую информацию для нахождения образа любой последовательности функции f.

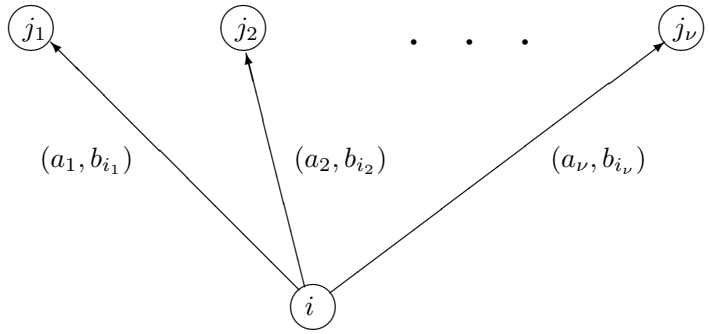


Рис. 4

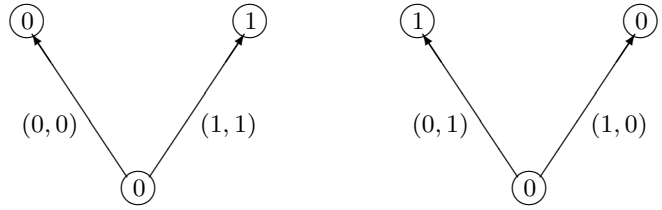


Рис. 5

На рис. 6 изображено усеченное дерево, построенное на основе информационного дерева рис. 3.

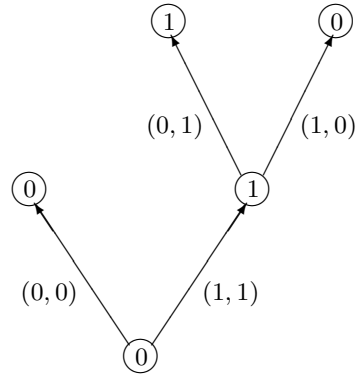


Рис. 6

Ограниченно-детерминированные функции удобно задавать *диаграммами переходов* (*диаграммы Мура*), которые получаются из усеченных деревьев отождествлением вершин с одинаковыми номерами. В результате получаем конечный ориентированный граф с r вершинами, которые занумерованы числами $0, 1, \ldots, r-1$, и $v \cdot r$ ребрами; при этом из каждой вершины графа выходит v ребер, которым приписаны пары

$$(a_1, b_{i_2}), (a_2, b_{i_2}), \ldots, (a_\nu, b_{i_\nu}),$$

где $\{a_1, \ldots, a_\nu\} = A$, $b_{i_1}, \ldots, b_{i_\nu} \in B$. Кроме того, вершина этого графа, соответствующая корню исходного информационного дерева T (при приведенном способе нумерации вершин она имеет номер 0), обычно помечается символом $*$.

На рис. 7 приведена диаграмма переходов для функции четности.

С диаграммами переходов можно связать две функции, F и G, $F : A \times Q \to B$ и $G : A \times Q \to Q$, которые называются функциями

выходов и переходов соответственно. Значения этих функций для всех $a_i \in A$, $q_i \in Q$ находятся в соответствии с рис. 8.

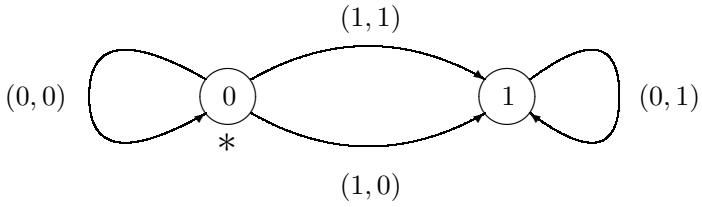


Рис. 7

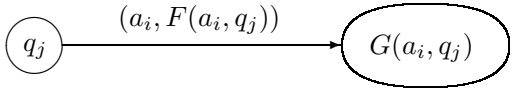


Рис. 8

В результате получаем способ задания о.-д. функций в виде таблиц, в $v \cdot r = |A| \cdot |Q|$ строках которых перечислены все пары (a_i, q_j) из $A \times Q$ и значения функций F и G на них (табл. 1).

Таблица 1

x	q	F	G
$\dots$	$\dots$	$\dots$	$\dots$
a_i	q_j	$F(a_i, q_j)$	$G(a_i, q_j)$
$\dots$	$\dots$	$\dots$	$\dots$

Значения функций F и G для функции четности приведены в табл. 2.

Таблица 2

x	q	F	G
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

О.-д. функции можно задавать также при помощи уравнений:

$$\begin{cases} y(t) = F(x(t), q(t)); \\ q(t+1) = G(x(t), q(t)); \\ q(1) = q_0, \end{cases}$$

где $x(t) \in A$, $y(t) \in B$, $q(t) \in Q$ при всех $t = 1, 2, \dots$; q_0 — номер вершины в диаграмме переходов, которая отмечена символом $*$, $q_0 \in Q$. Эти уравнения называются *каноническими уравнениями* о.-д. функции f с начальным состоянием q_0.

Легко видеть, что по каноническим уравнениям, которые задают о.-д. функцию f, можно получить и все другие перечисленные выше способы задания этой функции.

Канонические уравнения для функций четности (a) и единичной задержки (b) имеют следующий вид:

(a)

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

(b)

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

Конечный автомат — это устройство, функционирующее в дискретные моменты времени $t = 1, 2, \dots$, имеющее вход, выход и конечное число состояний $q_1, \dots, q_\lambda$. В момент времени t автомат находится в состоянии $q(t) \in Q = \{q_1, \dots, q_\lambda\}$, на его вход подается

символ $x(t)$ из конечного множества $A = \{a_1, \dots, a_\nu\}$, а на выходе возникает символ $y(t)$ из конечного множества $B = \{b_1, \dots, b_\mu\}$ (рис. 9).

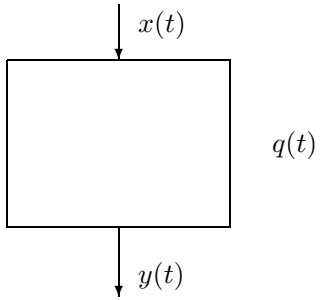


Рис. 9

При этом входной символ $x(t)$ и состояние автомата $q(t)$ однозначно определяют выходной символ $y(t)$ в момент времени t и состояние $q(t+1)$ автомата в следующий момент времени:

$$\begin{cases} y(t) = F(x(t), q(t)); \\ q(t+1) = G(x(t), q(t)), \end{cases}$$

$t = 1, 2, \dots$. Функции $F : A \times Q \to B$ и $G : A \times Q \to Q$ называются функциями выходов и переходов соответственно; множества A, B и Q называются соответственно входным алфавитом, выходным алфавитом и алфавитом состояний. Автомат называется *инициальным*, если задано его состояние $q(1)$ в начальный момент времени $t = 1,\ q(1) \in Q$.

Таким образом, конечный автомат полностью задается системой $V = (A, B, Q, F, G)$, где $A,\ B$ и Q — конечные множества входных символов, выходных символов и символов состояний соответственно; F — функция, определенная на множестве $A \times Q$ и принимающая значения из B, а G — функция, определенная на $A \times Q$ и принимающая значения из Q. Эта система V также называется конечным автоматом. Инициальный автомат V с начальным состоянием $q_0 \in Q$ обозначается через V_{q_0}.

Легко видеть, что каждый конечный инициальный автомат V_{q_0} вычисляет некоторую функцию, определенную на множестве A^∞ и принимающую значения из множества B^∞. Эта функция называется *автоматной* и обозначается через $f_{V_{q_0}}$; *состояниями* этой функции называются состояния автомата V_{q_0}. Более точно, функция $f(x)$, определенная на множестве A^∞ и принимающая значения из B^∞, называется *автоматной*, если существует конечный инициальный автомат, вычисляющий эту функцию, т. е. перерабатывающий любую входную последовательность α из A^∞ в выходную последовательность β из B^∞, такую, что $\beta = f(\alpha)$.

Легко видеть, что для каждой о.-д. функции веса r существует конечный инициальный автомат с r состояниями, вычисляющий эту функцию, и, наоборот, каждый инициальный конечный автомат с λ состояниями вычисляет некоторую о.-д. функцию веса r, где $r \leqslant \lambda$. Тем самым функция является автоматной тогда и только тогда, когда она ограниченно-детерминированная.

Отметим, что конечные автоматы (как и о.-д. функции) можно задавать при помощи таблиц, диаграмм перехода, информационных деревьев и их конечных фрагментов.

Пусть $\alpha = (\alpha(1), \alpha(2), \dots)$ — некоторая последовательность из A^∞. Натуральное число d называется *периодом* последовательности α, если существует такое N, что для любого $t \geqslant N$ выполняется равенство $\alpha(t+d) = \alpha(t)$. Последовательность называется *периодической*, если для нее существует хотя бы один период.

Заметим, что если d является периодом последовательности α, то периодами этой последовательности являются также все числа, кратные d. Поскольку из всех периодов периодической последовательности α можно выбрать минимальный период d_0, то все периоды последовательности кратны d_0.

Лемма (о периодической последовательности)**.** *Конечный инициальный автомат с λ состояниями преобразует периодическую последовательность с периодом d в периодическую последовательность с периодом $\lambda_1 \cdot d$, где λ_1 — натуральное число, $\lambda_1 \leqslant \lambda$.*

Доказательство. Пусть $V_{q_0} = \{A, B, Q, F, G\}$ — инициальный автомат, $|Q| = \lambda,\ q_0 \in Q,\ f_{V_{q_0}}(x)$ — автоматная функция, вычисляемая автоматом $V_{q_0},\ \alpha = (\alpha(1), \alpha(2), \dots)$ — периодическая последовательность из A^∞ с периодом d, а $\beta = f_{V_{q_0}}(\alpha) = (\beta(1), \beta(2), \dots)$ — последовательность из B^∞.

Так как d — период последовательности α, то существует такое N, что для любого $t \geqslant N$ выполняется равенство $\alpha(t) = \alpha(t+d)$. Поэтому

$$\alpha(N) = \alpha(N+d) = \alpha(n+2d) = \cdots = \alpha(N+\lambda d).$$

Рассмотрим $\alpha + 1$ состояний автомата:

$$q(N), q(N+d), \ldots, q(N+\lambda d).$$

Так как $|Q| = \lambda$, то среди них найдутся по крайней мере два одинаковых. То есть существуют такие i, j, $0 \leqslant i < j \leqslant \lambda$, что $q(N+id) = q(N+jd)$. Положим $\lambda_1 = j - i$.

Покажем индукцией по t, что для любого $t \geqslant N + id$ выполняются равенства

$$\beta(t) = \beta(t+\lambda_1 d), \quad q(t) = q(t+\lambda_1 d). \qquad (*)$$

Пусть $t = N + id$. Тогда

$$q(t) = q(N+id) = q(N+jd) = q(t+\lambda_1 d),$$

$$\beta(t) = F(\alpha(t), q(t)) = F(\alpha(t+\lambda_1 d), g(t+\lambda_1 d)) = \beta(t+\lambda_1 d).$$

Пусть равенства $(*)$ справедливы для некоторого $t \geqslant N + it$. Тогда

$$q(t+1) = G(\alpha(t), q(t)) =$$

$$= G(\alpha(t+\lambda_1 d), q(t+\lambda_1 d)) = q(t+1+\lambda_1 d),$$

$$\beta(t+1) = F(\alpha(t+1), q(t+1)) =$$

$$= F(\alpha(t+1+\lambda_1 d), q(t+1+\lambda_1 d)) = \beta(t+1+\lambda_1 d).$$

Таким образом, последовательность β является периодической с периодом $\lambda_1 d$.

Из леммы следует, что конечный автомат с λ состояниями может преобразовывать периодическую последовательность только в периодическую, у которой длина минимального периода может увеличиться не более чем в λ раз.

Заметим, что даже если число d — минимальный период входной последовательности, то минимальный период выходной последовательности не обязательно равен $\lambda_1 d$. В качестве примера достаточно рассмотреть автомат с λ состояниями, который любую входную последовательность преобразует в последовательность $(b_1, b_1, \ldots) \in B^\infty$.

Обозначим через A_k множество периодических последовательностей из A^∞, у которых минимальные периоды не имеют простых делителей, не превосходящих k, $k \in \mathbb{N}$. Например, множество A_2 состоит из последовательностей, минимальные периоды которых принадлежат множеству $\{2^n,\ n = 0, 1, \ldots\}$, а A_6 — из последовательностей с минимальными периодами из множества $\{2^n 3^l,\ n, l = 0, 1, \ldots\}$.

Из леммы о периодической последовательности получаем

Следствие. *Конечный инициальный автомат с λ состояниями при $\lambda \leqslant k$ преобразует последовательности из A_k в последовательности из A_k.*

Лекция № 10

Обобщим введенное нами ранее понятие схемы из функциональных элементов.

Пусть $\mathcal{F} = \{f_1^{(n_1)}(x_1, \ldots, x_{n_1}), \ldots, f_k^{(n_k)}(x_1, \ldots, x_{n_k})\}$ — конечное множество булевых функций. *Схема из функциональных элементов в базисе* $\mathcal{F}$ — это конечный ориентированный граф без ориентированных циклов, в каждую вершину которого либо не входит ни одно ребро, либо входит n_i ребер, занумерованных числами $1, 2, \ldots, n_i$, $1 \leqslant i \leqslant k$, и вершинам которого приписаны символы в соответствии со следующими правилами:

a) если в вершину не входят ребра, то этой вершине приписывается символ некоторой переменной;

b) если в вершину входит n_i ребер (занумерованных числами $1, 2, \ldots, n_i$), то этой вершине приписывается символ $f_i^{(n_i)}$;

c) кроме того, некоторым вершинам приписывается символ $*$.

Пример схемы из функциональных элементов в базисе $\{f_1^{(3)}(x_1, x_2, x_3), f_2^{(1)}(x_1)\}$ изображен на рис. 1.

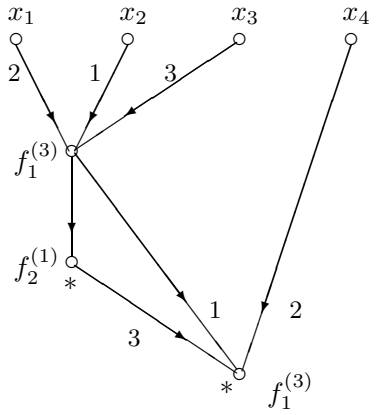


Рис. 1

После этого каждой вершине схемы можно поставить в соответствие некоторую булеву функцию по следующим правилам:

1) если вершине была приписана переменная, то ей ставится в соответствие функция, равная этой переменной;

2) если вершине v был приписан символ $f_i^{(n_i)}$, а вершинам $w_1, \ldots, w_{n_i}$, из которых выходят входящие в вершину v ребра, занумерованные числами $1, 2, \ldots, n_i$ соответственно, уже поставлены в соответствие функции $\varphi_1, \ldots, \varphi_{n_i}$, то вершине v сопоставляется функция $f_i^{(n_i)}(\varphi_1, \ldots, \varphi_{n_i})$ (рис. 2).

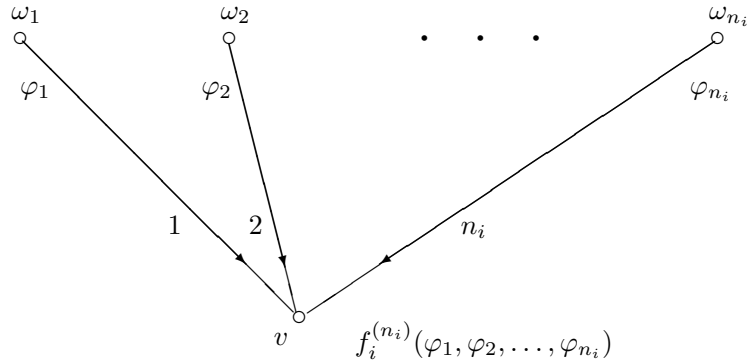


Рис. 2

Этот процесс в конце концов поставит в соответствие каждой вершине схемы некоторую функцию алгебры логики от переменных, приписанных вершинам схемы.

Функции, поставленные в соответствие вершинам, помеченным символом $*$, по определению *реализуются* этой схемой.

Легко видеть, что схема, изображенная на рис. 1, реализует функции $\varphi_1(x_1, x_2, x_3)$ и $\varphi_2(x_1, x_2, x_3, x_4)$, где

$$\varphi_1(x_1, x_2, x_3) = f_2^{(1)}(f_1^{(3)}(x_1, x_2, x_3)),$$
$$\varphi_2(x_1, x_2, x_3, x_4) = f_1^{(3)}(f_1^{(3)}(x_1, x_2, x_3), x_4, \varphi_1(x_1, x_2, x_3)).$$

Вершины, которым приписаны переменные, называются *входами* схемы; вершины, которым приписаны символы $f_i^{(n_i)}$, где $i = 1, \ldots, k$, называются *элементами* (*функциональными элементами*); вершины, которым приписан символ $*$, — *выходами* схемы.

Наряду с введенным ранее понятием конечного автомата (с одним входом) можно рассмотреть автоматы с несколькими входами. А именно будем считать, что конечный автомат имеет n

входов, занумерованных числами $1, 2, \ldots, n$. В момент времени t, $t = 1, 2, \ldots$, автомат находится в состоянии $q(t)$ из алфавита состояний $Q = \{q_1, \ldots, q_\lambda\}$, на его входы $1, 2, \ldots, n$ подаются соответственно символы $x_1(t), x_2(t), \ldots, x_n(t)$ из входного алфавита $A = \{a_1, \ldots, a_\nu\}$, а на выходе возникает символ $y(t)$ из выходного алфавита $B = \{b_1, \ldots, b_\mu\}$ (рис. 3).

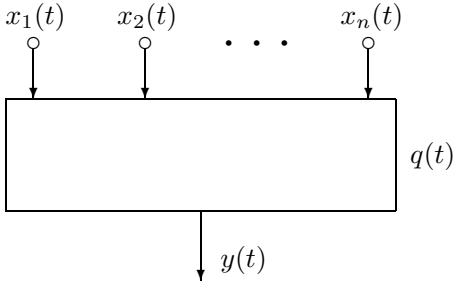


Рис. 3

При этом входные символы $x_1(t), x_2(t), \ldots, x_n(t)$ и состояние автомата $q(t)$ однозначно определяют выходной символ $y(t)$ и состояние автомата в следующий момент времени:

$$\begin{cases} y(t) = F(x_1(t), x_2(t), \ldots, x_n(t), q(t)); \\ q(t+1) = G(x_1(t), x_2(t), \ldots, x_n(t), q(t)), \end{cases}$$

$t = 1, 2, \ldots$. Функции $F : A^n \times Q \to B$ и $G : A^n \times Q \to Q$ так же, как и ранее, называются функциями выходов и переходов соответственно. Автомат называется инициальным, если задано его состояние $q(1)$ в начальный момент времени.

Пусть $\alpha_1 = (\alpha_1(1), \alpha_1(2), \ldots)$, $\alpha_2 = (\alpha_2(1), \alpha_2(2), \ldots), \ldots,$ $\alpha_n = (\alpha_n(1), \alpha_n(2), \ldots)$ — последовательности из A^∞, которые подаются соответственно на входы $1, 2, \ldots, n$ автомата с n входами. Можно считать, что автомат перерабатывает упорядоченный набор входных последовательностей $\alpha_1, \alpha_2, \ldots, \alpha_n$ в выходную последовательность $\beta = (\beta(1), \beta(2), \ldots)$ из B^∞. Поэтому аналогично понятию автоматной функции одной переменной можно определить понятие автоматной функции от нескольких переменных. А именно функция $f(x_1, \ldots, x_n)$, определенная на множестве $(A^\infty)^n$ и принимающая значения из B^∞, называется *автоматной*, если существует конечный инициальный автомат с n входами, вычисляющий эту функцию, т. е. перерабатывающий любой набор последовательностей $\alpha_1, \ldots, \alpha_n$ из A^∞ в выходную последовательность β из B^∞, такую, что $\beta = f(\alpha_1, \ldots, \alpha_n)$. *Состояниями* автоматной функции будем называть состояния автомата, вычисляющего эту функцию.

Отметим, что описанный выше автомат с n входами можно рассматривать так же, как автомат с одним входом, на который в каждый момент времени t, $t = 1, 2, \ldots$, подается символ $x(t) = (x_1(t), x_2(t), \ldots, x_n(t)) \in A^n$. Такой автомат будет вычислять некоторую автоматную функцию $f(x)$, определенную на множестве $(A^n)^\infty$ и принимающую значения из B^∞.

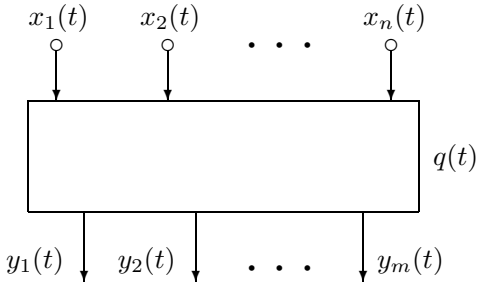


Рис. 4

Аналогичным образом можно рассмотреть автоматы с n входами и m выходами, занумерованными числами $1, 2, \ldots, m$. В момент времени t автомат находится в состоянии $q(t) \in Q$, на его входы подаются символы $x_1(t), x_2(t), \ldots, x_n(t) \in A$, а на m выходах $1, 2, \ldots, m$ выдаются соответственно символы $y_1(t), y_2(t), \ldots, y_m(t)$ из B (рис. 4). При этом $x_1(t), x_2(t), \ldots, x_n(t)$ и $q(t)$ однозначно определяют выходные символы $y_1(t)$, $y_2(t), \ldots,$ $y_m(t)$ и состояние $q(t+1)$ в следующий момент времени:

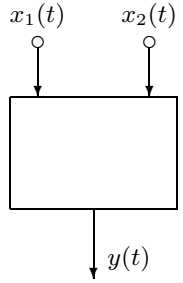
$$\begin{cases} y_1(t) = F_1(x_1(t), x_2(t), \ldots, x_n(t), q(t)); \\ \ldots \\ y_m(t) = F_m(x_1(t), x_2(t), \ldots, x_n(t), q(t)); \\ q(t+1) = G(x_1(t), x_2(t), \ldots, x_n(t), q(t)), \end{cases}$$

$t = 1, 2, \ldots$, где $F_i : A^n \times Q \to B$, $i = 1, \ldots, m$, и $G : A^n \times Q \to Q$.

Автоматы с n входами и m выходами вычисляют упорядоченный набор автоматных функций.

Пример. Рассмотрим сумматор, имеющий два входа, на которые подаются последовательности из нулей и единиц $x_1 = (x_1(1), x_1(2), \dots)$ и $x_2 = (x_2(1), x_2(2), \dots)$, и один выход. На выходе сумматора выдается последовательность из нулей и единиц $y = (y(1), y(2), \dots)$, которая получается путем обычного сложения в двоичной системе счисления последовательностей x_1 и x_2 как целых чисел, имеющих бесконечное число разрядов:

$$\begin{array}{cccccc} & x_1(1) & x_1(2) & x_1(3) & \dots \\ + & & & & & \\ & x_2(1) & x_2(2) & x_2(3) & \dots \\ \hline & y(1) & y(2) & y(3) & \dots \end{array}$$



В качестве такого сумматора можно взять инициальный автомат $V_{q_0} = (A, B, Q, F, G)$ с двумя входами, у которого $A = B = \{0, 1\}$, $Q = \{0, 1\}$, а функционирование задается следующими уравнениями:

$$\begin{cases} y(t) = x_1(t) + x_2(t) + q(t); \\ q(t+1) = x_1(t)(x_2(t) \vee q(t)) \vee x_2(t)q(t); \\ q(1) = 0. \end{cases}$$

Обозначим через P_A множество всех автоматных функций (от любого числа переменных), для которых входной и выходной алфавиты равны A (т. е. $B = A$).

Аналогично понятию схемы из функциональных элементов в базисе из функций алгебры логики можно ввести понятие схемы из автоматных элементов в базисе

$$F = \{f_1^{(n_1)}(x_1, \dots, x_{n_1}), \dots, f_k^{(n_k)}(x_1, \dots, x_{n_k})\} \subseteq P_A.$$

Для этого достаточно точно так же, как мы это делали для схем из функциональных элементов, рассмотреть конечный ориентированный граф без ориентированных циклов с занумерованными ребрами, приписать каждой вершине графа либо символ переменной

(если в эту вершину не входит ни одно ребро), либо символ автоматной функции $f_i^{(n_i)}$ (если в вершину входит n_i ребер) и, кроме того, пометить некоторые вершины символом $*$.

Полученный объект называется *схемой из автоматных элементов в базисе* F. Вершины, которым приписаны переменные, называются *входами* схемы; вершины, которым приписаны символы $f_i^{(n_i)}$, $i = 1, \dots, k$, называются *элементами* (*автоматными элементами*); вершины, которым приписан символ $*$, — *выходами* схемы.

После этого (так же, как и ранее) каждой вершине схемы из автоматных элементов можно сопоставить некоторую автоматную функцию из P_A. Функции, поставленные в соответствие вершинам, помеченным символом $*$, по определению *реализуются* этой схемой.

Заметим, что в силу определения автоматных функций любой элемент схемы из автоматных элементов можно рассматривать как конечный инициальный автомат с несколькими входами, каждый из которых присоединен либо ко входу схемы, либо к выходу другого элемента схемы. Поэтому всю схему из автоматных элементов можно рассматривать как некоторый "сложный" инициальный автомат с несколькими входами и выходами. Состояние этого автомата полностью определяется состояниями всех элементов схемы, и поэтому он должен иметь не более чем конечное число различных состояний. Кроме того, на каждом выходе данный автомат вычисляет функцию, которая совпадает с функцией, сопоставленной этому выходу схемы.

Таким образом, все функции, которые реализуются схемами из автоматных элементов в базисах из P_A, являются автоматными, т. е. принадлежат P_A.

Пусть $\mathcal{F}$ — некоторая система функций из P_A. Замыканием системы $\mathcal{F}$ называется множество $\Sigma(\mathcal{F})$, состоящее из всех функций, которые реализуются схемами из автоматных элементов в базисе $\mathcal{F}$. Система $\mathcal{F}$ называется полной, если $\Sigma(\mathcal{F}) = P_A$.

Теорема. *Пусть* $|A| \geqslant 2$. *Тогда в* P_A *не существует конечных полных систем автоматных функций.*

Докажем сначала следующее утверждение.

Лемма. *Пусть* k *— натуральное число,* $\mathcal{F}$ *— конечная система функций из* P_A, *каждая функция которой имеет не более* k *состояний,* S *— схема из автоматных элементов в базисе*

$\mathcal{F}$, $f(x_1,\dots,x_n)$ — автоматная функция, реализуемая схемой S, а $\alpha_1,\dots,\alpha_n$ — последовательности из A_k. Тогда последовательность $\beta = f(\alpha_1,\dots,\alpha_n)$ принадлежит A_k.

Доказательство. Воспользуемся индукцией по числу N элементов схемы S. Если $N = 0$, то утверждение леммы очевидно. Предположим, что утверждение леммы справедливо для всех схем, имеющих не более $N-1$ элементов.

Рассмотрим схему S в базисе $\mathcal{F}$, состоящую из N элементов. Возьмем произвольный выход схемы S. Если он совпадает со входом, то утверждение леммы очевидно. В противном случае этот выход является выходом некоторого элемента $g \in \mathcal{F}$ схемы S. Обозначим через h и λ число входов и число состояний элемента g соответственно. Пусть i-й вход элемента g присоединен к вершине v_i схемы S, $i = 1,\dots,h$ (рис. 5).

Тогда вершины $v_1,\dots,v_h$ можно рассматривать как выходы некоторой подсхемы S_1 схемы S, которая не содержит элемент g и на выходах $v_1,\dots,v_h$ реализует некоторые автоматные функции $\varphi_1(x_1,\dots,x_n),\dots,\varphi_h(x_1,\dots,x_n)$ соответственно.

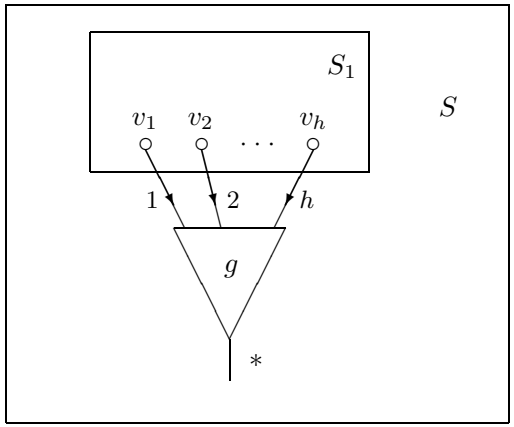


Рис. 5

Пусть $\alpha_1,\dots,\alpha_n$ — произвольные последовательности из A_k. Рассмотрим $\gamma_1 = \varphi_1(\alpha_1,\dots,\alpha_n),\dots,\ \gamma_h = \varphi_h(\alpha_1,\dots,\alpha_n)$. Так как S_1 содержит менее N элементов и является схемой в базисе $\mathcal{F}$, то по

предположению индукции последовательности $\gamma_1,\dots,\gamma_h$ принадлежат A_k. Обозначим через $d_1,\dots,d_h$ минимальные периоды последовательностей $\gamma_1,\dots,\gamma_h$ соответственно. Заметим, что элемент g можно рассматривать как автомат с одним входом, на который поступают бесконечные последовательности $\gamma = (\gamma(1),\gamma(2),\dots)$ из $(A^h)^\infty$. То есть в момент времени t на вход g поступает символ $\gamma(t) = (\gamma_1(t),\gamma_2(t),\dots,\gamma_h(t))$ из A^h. Легко видеть, что γ — это периодическая последовательность, у которой минимальный период $d = \text{НОК}(d_1,\dots,d_h)$. Поэтому в силу леммы о периодической последовательности на выходе g выдается периодическая последовательность $\beta = g(\gamma)$ из A^∞ с периодом $\lambda_1 d$, где $\lambda_1 \leqslant \lambda \leqslant k$. Так как числа $d_1,\dots,d_h$ не имеют простых делителей, превышающих k, то d также не имеет простых делителей, превышающих k. Поэтому и $\lambda_1 d$ не имеет простых делителей, превосходящих k. Таким образом, $\beta \in A_k$.

Доказательство теоремы. Предположим, что в P_A существует конечная полная система $\mathcal{F}$. Обозначим через k максимальное число состояний у функции системы $\mathcal{F}$.

Пусть p — простое число, такое, что $p > k$, а β — периодическая последовательность из A^∞ с периодом p следующего вида:

$$\beta = (\underbrace{a_1,\dots,a_1}_{p-1}, a_2, \underbrace{a_1,\dots,a_1}_{p-1}, a_2, a_1,\dots),$$

где $a_1, a_2 \in A$, $a_1 \neq a_2$. Очевидно, что $\beta \notin A_k$. Рассмотрим функцию $f(x)$, такую, что $f : A^\infty \to A^\infty$, и для любой последовательности α из A^∞ выполняется равенство $f(\alpha) = \beta$. Легко видеть, что $f \in P_A$.

Так как по предположению $\mathcal{F}$ — полная система, то существует схема S в базисе $\mathcal{F}$, реализующая функцию $f(x)$. Рассмотрим последовательность $\gamma = (a_1, a_1,\dots) \in A^\infty$, состоящую только из букв a_1, и подадим ее на вход схемы S. Так как минимальный период последовательности γ равен 1, то $\gamma \in A_k$. Поэтому в силу предыдущей леммы на выходах схемы S будут выдаваться последовательности из A_k, т. е. $f(\gamma) \in A_k$. Но по выбору функции f имеем $f(\gamma) = \beta \notin A_k$. Полученное противоречие показывает, что в P_A не существует конечных полных систем автоматных функций.

Теорема доказана.

Таким образом, эта теорема дает отрицательный ответ на вопрос о существовании конечных полных систем автоматных функ-

ций. Однако если расширить некоторым образом возможности при построении схем из автоматных элементов, то такие системы можно построить.

Схемы из функциональных элементов и элементов задержки. Будем реализовывать автоматные функции из множества P_E, где $E = \{0, 1\}$, при помощи функциональных элементов дизъюнкции, конъюнкции и отрицания и элемента единичной задержки (рис. 6).

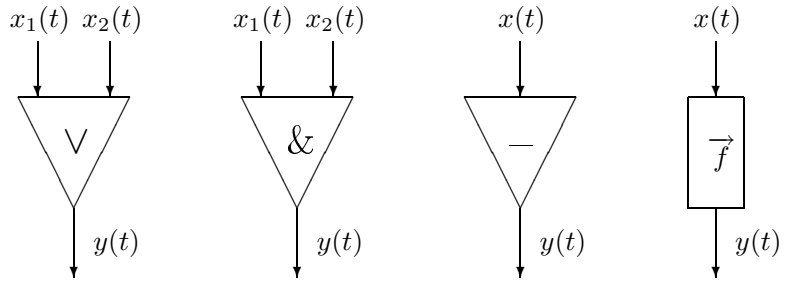


Рис. 6

Отметим, что функциональные элементы можно рассматривать как частный случай автоматов, а точнее, как автоматы с одним состоянием. Такие автоматы называются автоматами без памяти. В частности, канонические уравнения для автоматов, соответствующих элементам дизъюнкции, конъюнкции и отрицания, имеют следующий вид:

$$\begin{cases} y(t) = x_1(t) \vee x_2(t); \\ q(t+1) = q(t); \\ q(1) = 0, \end{cases}$$

$$\begin{cases} y(t) = x_1(t) \& x_2(t); \\ q(t+1) = q(t); \\ q(1) = 0, \end{cases}$$

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

Будем обозначать автоматные функции из P_E, которые вычисляются этими автоматами, через $f_\vee(x_1, x_2)$, $f_\&(x_1, x_2)$ и $f_-(x)$ соответственно. Канонические уравнения элемента единичной задержки имеют вид

$$\begin{cases} y(t) = q(t); \\ q(t+1) = x(t); \\ q(1) = 0, \end{cases}$$

соответствующая автоматная функция обозначается через $\vec{f}(x)$.

Пусть $f(x_1, \ldots, x_n)$ — произвольная автоматная функция из P_E, а $V_{q_0} = (A, B, Q, F, G)$ — инициальный автомат, вычисляющий функцию f, где $A = E^n = \{0, 1\}^n$, $B = \{0, 1\}$, $Q = \{q_1, \ldots, q_\lambda\}$, $q_0 \in Q$, $F : \{0, 1\}^n \times Q \to \{0, 1\}$, $G : \{0, 1\}^n \times Q \to Q$. Канонические уравнения этого автомата имеют вид

$$\begin{cases} y(t) = F(x_1(t), \ldots, x_n(t), q(t)); \\ q(t+1) = G(x_1(t), \ldots, x_n(t), q(t)); \\ q(1) = q_0. \end{cases}$$

Положим[1] $l =]\log_2 \lambda[$. Занумеруем состояния $q_1, \ldots, q_\lambda$ наборами из 0 и 1 длины l, причем начальному состоянию q_0 сопоставим набор $(0, \ldots, 0)$. Рассмотрим теперь новые функции, определенные на наборах из нулей и единиц:

$$\begin{cases} y(t) = F^1(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_l(t)); \\ (q_1(t+1), \ldots, q_l(t+1)) = \tilde{G}^1(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_l(t)); \\ (q_1(1), \ldots, q_l(1)) = (0, \ldots, 0), \end{cases}$$

где функция F^1 и компоненты $G^1_1, \ldots, G^1_l$ вектор-функции $\tilde{G}^1 = (G^1_1, \ldots, G^1_l)$ определены на некотором подмножестве множества E^{n+l} всех наборов из нулей и единиц длины $n + l$ и принимают значения из $\{0, 1\}$, т. е. являются частичными булевыми функциями. В покомпонентной записи эти уравнения имеют следующий вид:

[1] Через $]a[$ обозначается наименьшее целое число, не меньшее a.

$$\begin{cases} y(t) = F^1(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_l(t)); \\ q_1(t+1) = G_1^1(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_l(t)); \\ \ldots \\ q_l(t+1)) = G_l^1(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_l(t)); \\ q_1(1) = 0; \\ \ldots \\ q_l(1) = 0. \end{cases}$$

Доопределим функции $F^1, G_1^1, \ldots, G_l^1$ до всюду определенных булевых функций $F^2, G_1^2, \ldots, G_l^2$ соответственно. Положим $\tilde{G}^2 = (G_1^2, \ldots, G_l^2)$.

Таким образом, мы построили инициальный автомат $V_{\tilde{q}_0}^2 = (\{0,1\}^n, \{0,1\}, \{0,1\}^l, F^2, \tilde{G}^2)$ с n входами, входным алфавитом $\{0,1\}^n$, выходным алфавитом $\{0,1\}$, алфавитом состояний $\{0,1\}^l$, функцией выходов $F^2 : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}$, вектор-функцией переходов $\tilde{G}^2 : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^l$ и начальным состоянием $\tilde{q}_0 = (0, \ldots, 0)$. Легко видеть, что этот автомат вычисляет функцию $f(x_1, \ldots, x_n)$.

Построим схему S из функциональных элементов в базисе $\{\vee, \&, \bar{}\}$ с $n + l$ входами, которым приписаны символы $x_1, \ldots, x_n, q_1, \ldots, q_l$, и $1 + l$ выходами $y, z_1, \ldots, z_l$; при этом на выходе y реализуется функция $F^2(x_1, \ldots, x_n, q_1, \ldots, q_l)$, а на выходах $z_1, \ldots, z_l$ — функции $G_1^2(x_1, \ldots, x_n, q_1, \ldots, q_l), \ldots, G_l^2(x_1, \ldots, x_n, q_1, \ldots, q_l)$ соответственно (рис. 7).

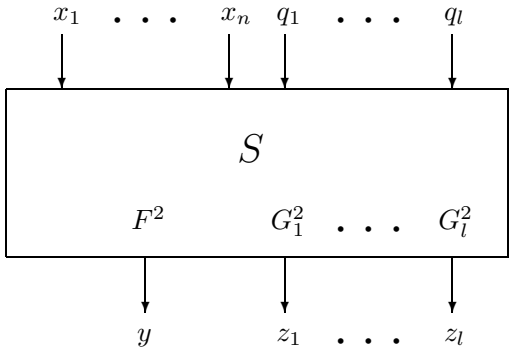


Рис. 7

Преобразуем схему S следующим образом. Возьмем l элементов единичной задержки. Присоединим вход i-го элемента задержки к выходу z_i, а выход — ко входу q_i схемы S, $i = 1, \ldots, l$. Кроме того, символы $x_1, \ldots, x_n$ заменим на $x_1(t), \ldots, x_n(t)$ соответственно, а символы функциональных элементов $\vee, \&, \bar{}$ — на символы $f_\&$, $f_\vee$ и f_- соответственно. В результате получим "схему" S_1 из автоматных элементов в базисе $\{f_\&, f_\vee, f_-, \vec{f}\}$ (рис. 8).

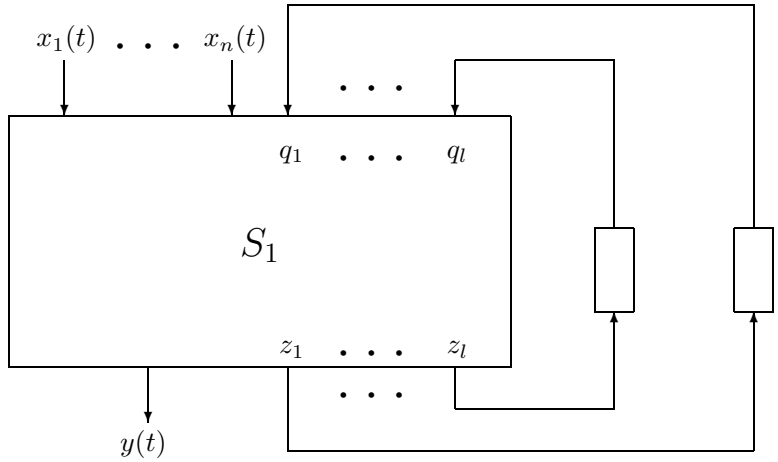


Рис. 8

Отметим, что при этом мы вышли за рамки данного ранее определения схемы из автоматных элементов, поскольку в S_1 возникли ориентированные циклы. Однако их наличие в "схеме" S_1 не приводит к противоречиям при ее функционировании, поскольку все эти циклы проходят через элементы задержки.

Операция построения в схемах ориентированных циклов, проходящих через элементы задержки, называется операцией *обратной связи*.

Индукцией по t нетрудно показать, что если в момент времени t подавать на входы схемы S_1 значения $x_1(t), \ldots, x_n(t)$ из $\{0,1\}$, то на ее выходе будет выдаваться значение $y(t)$ из $\{0,1\}$, которое вычисляется в соответствии с каноническими уравнениями автомата $V_{\tilde{q}_0}^2$.

Тем самым схема S_1 реализует автоматную функцию $f(x_1,\dots,x_n)$. Таким образом, имеет место

Теорема. *Любую автоматную функцию из P_E можно реализовать схемой из автоматных элементов в базисе $\{f_\&, f_\vee, f_-, \vec{f}\}$ с использованием операции обратной связи.*

Заметим, что любая схема S, построенная из функциональных элементов с использованием операции обратной связи, реализует некоторую автоматную функцию из P_E, поскольку состояние схемы полностью определяется состояниями элементов задержки, а значит, схема S имеет лишь конечное число различных состояний.

Таким образом, замыкание системы $\{f_\&, f_\vee, f_-, \vec{f}\}$ (с использованием операции обратной связи) совпадает с множеством P_E всех автоматных функций.

Рассмотрим теперь произвольную автоматную функцию $f(x)$, $f : A^\infty \to B^\infty$. Пусть $V_{q_0}(A,B,Q,F,G)$ — инициальный автомат, вычисляющий функцию f, где $A=\{a_1,\dots,a_\nu\}$ — входной алфавит, $B=\{b_1,\dots,b_\mu\}$ — выходной алфавит, $Q=\{q_1,\dots,q_\lambda\}$ — алфавит состояний, $F : A\times Q\to B$ и $G : A\times Q\to Q$ — функции выходов и переходов соответственно, а q_0 — начальное состояние автомата V_{q_0}. Канонические уравнения этого автомата имеют вид

$$\begin{cases} y(t) = F(x(t), q(t)); \\ q(t+1) = G(x(t), q(t)); \\ q(1) = q_0. \end{cases}$$

Положим

$$n =]\log_2 \nu[, \qquad m =]\log_2 \mu[, \qquad l =]\log_2 \lambda[.$$

Занумеруем буквы алфавитов A, B и Q наборами из нулей и единиц длины n, m и l соответственно, причем состоянию q_0 сопоставим набор $(0,\dots,0)$. Определим новые функции:

$$\begin{cases} (y_1(t),\dots,y_m(t)) = \tilde{F}(x_1(t),\dots,x_n(t),q_1(t),\dots,q_l(t)), \\ (q_1(t+1),\dots,q_l(t+1)) = \tilde{G}(x_1(t),\dots,x_n(t),q_1(t),\dots,q_l(t)), \\ (q_1(1),\dots,q_l(1)) = (0,0,\dots,0), \end{cases}$$

где $\tilde{F}=(F_1,\dots,F_m)$ и $\tilde{G}=(G_1,\dots,G_l)$ — вектор-функции, компоненты которых определены на некотором подмножестве множества E^{n+l} и которые принимают значения из $E=\{0,1\}$.

Доопределим функции $\tilde{F}$ и $\tilde{G}$ до всюду определенных вектор-функций $\tilde{F}^1$ и $\tilde{G}^1$ соответственно.

В результате мы построили инициальный автомат

$$V^1_{\tilde{q}_0} = (\{0,1\}^n, \{0,1\}^m, \{0,1\}^l, \tilde{F}^1, \tilde{G}^1)$$

с n входами и m выходами, входным алфавитом $\{0,1\}^n$, выходным алфавитом $\{0,1\}^m$, алфавитом состояний $\{0,1\}^l$, вектор-функцией выходов

$$\tilde{F}^1 : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^m,$$

вектор-функцией переходов

$$\tilde{G}^1 : \{0,1\}^n \times \{0,1\}^l \to \{0,1\}^l$$

и начальным состоянием $\tilde{q}_0 = (0,\dots,0)$. Этот автомат вычисляет некоторую систему автоматных функций $f_1(x_1,\dots,x_n),\dots,f_m(x_1,\dots,x_n)$ из P_E.

В момент времени t на выходах автомата $V^1_{\tilde{q}_0}$ выдается набор $(y_1(t),\dots,y_m(t))$ из нулей и единиц, по которому однозначно декодируется значение $y(t)\in B$ выхода автомата V_{q_0} в тот же момент времени, $t=1,2,\dots$. Таким образом, автомат $V^1_{\tilde{q}_0}$ моделирует функционирование автомата V_{q_0}.

Поскольку автомат $V^1_{\tilde{q}_0}$ вычисляет систему автоматных функций из P_E, то в силу теоремы существует схема S в базисе $\{f_\&, f_\vee, f_-, \vec{f}\}$, которая реализует эти функции. Поэтому для любой автоматной функции $f(x)$ можно построить схему из автоматных элементов в базисе $\{f_\&, f_\vee, f_-, \vec{f}\}$ с использованием операции обратной связи, реализующую систему автоматных функций $f_1,\dots,f_m$ из P_E, значения которых однозначно определяют значение исходной функции f.

АЛГОРИТМЫ И ВЫЧИСЛИМЫЕ ФУНКЦИИ

Лекция № 11

Важную роль в математике играет понятие *алгоритма* — четко описанной и приводящей к результату процедуры преобразования информации. В силу большого разнообразия существующих вычислительных процедур строгое математическое определение этого понятия представляет сложную задачу.

В тех случаях, когда для решения задач алгоритм в каком-то виде удается найти, этим обычно и ограничиваются. Однако для некоторых задач эффективных способов вычисления долгое время найти не удавалось (к их числу относится, например, разрешимость в целых числах диофантовых уравнений). Это послужило поводом предположить, что таких алгоритмов вообще не существует.

Пользуясь интуитивным понятием алгоритма, этот факт установить невозможно. В связи с этим были предприняты попытки формализовать понятие алгоритма. В середине 30-х гг. XX в. были предложены различные формальные определения алгоритма (А. Тьюринг, А. Чёрч, Э. Пост, А. А. Марков и др.). Но все они оказались эквивалентными друг другу. Поэтому было высказано предположение, что к данным определениям может быть сведено любое корректное определение алгоритма, т. е. каждое из них описывает класс всех возможных эффективных вычислений. Это позволяет дать строгую математическую постановку вопросу об алгоритмической разрешимости той или иной проблемы. Мы дадим определение алгоритма при помощи машины Тьюринга.

Машины Тьюринга. Машина Тьюринга — это математическая модель вычислительного устройства.

Каждая машина Тьюринга состоит из бесконечной в обе стороны ленты, разбитой на ячейки, и читающей и пишущей головки, которая может перемещаться вдоль ленты. Машина работает во времени, которое предполагается дискретным (моменты времени занумерованы числами $1, 2, \dots$). В каждой ячейке ленты в каждый момент времени записан символ из некоторого конечного алфавита $A = \{a_0, a_1, \dots, a_k\}$ (будем также называть A *входным алфавитом* машины). Будем предполагать, что среди букв алфавита A имеется

пустой символ a_0 (этот символ обозначается также через 0). Будем считать, что на ленте в начальный момент времени записано конечное число непустых символов.

Читающая и пишущая головка имеет конечное множество состояний $Q = \{q_0, q_1, \dots, q_n\}$, $Q \cap A = \emptyset$. В каждый момент времени t она находится в некотором состоянии $q(t) \in Q$ и воспринимает содержимое $x(t) \in A$ одной из ячеек ленты. В зависимости от состояния $q(t)$ и буквы $x(t)$ она записывает новую букву $y(t) \in A$ в ту же ячейку, к следующему моменту времени $t+1$ переходит в новое состояние $q(t+1) \in Q$ и сдвигается по ленте — либо на одну ячейку влево, либо на одну ячейку вправо, либо остается на месте. Среди состояний машины имеется заключительное состояние $q_0 \in Q$, попав в которое машина перестает работать.

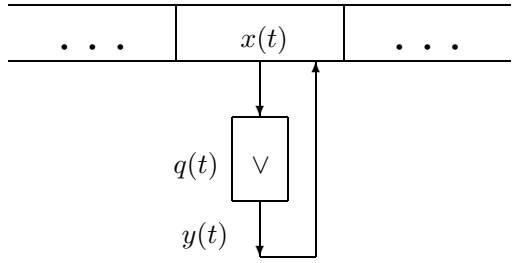


Рис. 1

Таким образом, читающая и пишущая головка представляет собой конечный автомат $V = (A, A, Q, F, G)$, у которого входным и выходным алфавитом служит $A = \{a_0, a_1, \dots, a_k\}$, множеством состояний является $Q = \{q_0, q_1, \dots, q_n\}$, $n \geqslant 1$, а $F : A \times Q \to A$ и $G : A \times Q \to Q$ — функции выходов и переходов соответственно. Кроме того, у этого автомата заданы заключительное состояние $q_0 \in Q$ и функция движения головки $D : A \times Q \to \{L, R, S\}$, где символы L, R и S обозначают сдвиг головки на одну ячейку влево, сдвиг на одну ячейку вправо и отсутствие движения соответственно (рис. 1).

В каждый момент времени на ленте записано некоторое слово, составленное из букв алфавита A, читающая и пишущая головка находится в некотором состоянии $q \in Q$ и обозревает некоторую ячейку ленты. Совокупность этих трех объектов будем называть

конфигурацией. Обычно будут рассматриваться конфигурации, в которых на ленте записано конечное число непустых символов. Такая конфигурация представлена на рис. 2, где изображена часть ленты, содержащая все непустые символы; под отмеченной обозреваемой ячейкой указывается состояние q головки; $a_{i_1}, \ldots, a_{i_r} \in A$. Можно считать в этом случае, что на ленте записано слово конечной длины в алфавите A, самая левая и самая правая буквы которого — непустые (либо на ленте записаны лишь пустые символы). Очевидно, что конфигурация машины в начальный момент времени полностью определяет ее работу. Как правило, в начальной конфигурации головка будет находиться в состоянии $q_1 \in Q$ и обозревать самую левую ячейку, занятую непустой буквой.

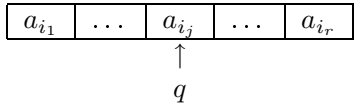


Рис. 2

Машина M называется *применимой* к начальной конфигурации (или к начальному слову), если она, работая в соответствии с установленными правилами, в некоторый момент времени придет в заключительное состояние. Результатом работы машины считается слово, записанное на ленте в заключительной конфигурации. Если же машина ни в какой момент времени не окажется в заключительном состоянии, то она называется *неприменимой* к начальной конфигурации (или к начальному слову); результат ее работы в этом случае неопределен.

Каждая машина полностью определяется конечным числом команд вида

$$qa \to q'a'd,$$

где q — состояние головки; a — буква в обозреваемой ячейке; $q' = G(a,q)$ — состояние головки в следующий момент времени; $a' = F(a,q)$ — буква, записываемая вместо буквы a в обозреваемую ячейку; $d = D(a,q)$ — сдвиг головки к следующему моменту времени: L, если происходит сдвиг влево; R, если вправо; S, если головка остается на месте. Совокупность всех команд машины называется ее *программой*. Отметим, что для всех q_i, a_j $(i = 1, \ldots, n;\ j = 0, \ldots, k)$ программа машины содержит в точности одну команду, которая начинается словом $q_i a_j$. Таким образом, программа машины с входным алфавитом $A = \{a_0, a_1, \ldots, a_k\}$ и множеством состояний $Q = \{q_0, q_1, \ldots, q_n\}$ содержит в точности $n(k+1)$ команд.

Пример. Пусть M_1 — машина Тьюринга со списком команд

$$q_i a_j \to q_i a_j S \ (i = 1, \ldots, n;\ j = 0, \ldots, k),$$

а M_2 — машина Тьюринга со списком команд

$$q_i a_j \to q_0 a_j S \ (i = 1, \ldots, n;\ j = 0, \ldots, k).$$

Очевидно, что машина M_1 не применима ни к какому слову, а машина M_2 применима к любому слову, записанному на ленте в начальный момент времени.

Для вычисления на машинах Тьюринга числовых функций используется следующее кодирование чисел (под числами понимаются целые неотрицательные числа $0, 1, 2, \ldots$, множество которых обозначается через $\mathbb{N}$). Число $m \in \mathbb{N}$ кодируется словом длины $m+1$, состоящим из одних единиц; набор $\tilde{\alpha} = (m_1, m_2, \ldots, m_n)$ кодируется словом $k(\tilde{\alpha})$ из нулей и единиц следующего вида:

$$k(\tilde{\alpha}) = \underbrace{1\ldots1}_{m_1+1}0\underbrace{1\ldots1}_{m_2+1}0\ldots0\underbrace{1\ldots1}_{m_n+1},$$

$m_1, m_2, \ldots, m_n \in N$; слово $k(\tilde{\alpha})$ называется *кодом* набора $\tilde{\alpha}$.

Пусть $f(x_1, \ldots, x_n)$ — частичная функция, определенная на некотором подмножестве $D(f)$ множества $\mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N} = \mathbb{N}^n$ и принимающая значения из множества $\mathbb{N}$; $D(f)$ называется областью определения функции f. Пусть M — некоторая машина Тьюринга. Машина M *вычисляет* функцию $f(x_1, \ldots, x_n)$, если для любого набора $\tilde{\alpha} = (m_1, m_2, \ldots, m_n)$ из $\mathbb{N}^n$ выполняются следующие условия:

a) если $\tilde{\alpha} \notin D(f)$, то M неприменима к слову $k(\tilde{\alpha})$;

b) если $\tilde{\alpha} \in D(f)$, то M применима к слову $k(\tilde{\alpha})$, и заключительная конфигурация имеет вид

$$0\underbrace{1\ldots1}_{m+1}0,$$
$$\uparrow$$
$$q_0$$

где $m = f(m_1, \ldots, m_n)$.

Частичная числовая функция называется *вычислимой* (по Тьюрингу), если существует машина Тьюринга, которая вычисляет эту функцию.

Отметим, что машина M_1 из первого вычисляет нигде не определенную функцию (считаем, что $0, 1 \in A$).

Пример. Рассмотрим машину M_3 с входным алфавитом $A = \{0, 1\}$, множеством состояний $Q = \{q_0, q_1, q_2\}$ и командами

$$q_1 1 \to q_1 1 R,$$
$$q_1 0 \to q_2 1 L,$$
$$q_2 1 \to q_2 1 L,$$
$$q_2 0 \to q_2 0 R.$$

Рассмотрим функционирование этой машины на следующей начальной конфигурации:

$$0\underbrace{1\ldots1}_{m+1}0\,.$$
$$\uparrow$$
$$q_1$$

Она сначала дойдет до конца массива из единиц, обнаружит нуль, заменит его на единицу, пойдет обратно и будет идти до тех пор, пока не найдет нуль, расположенный в начале числа, после чего остановится. Легко видеть, что машина M_3 вычисляет функцию $f(x) = x + 1$.

Если взять два числа m_1 и m_2 и рассмотреть функционирование машины M_3 при начальной конфигурации

$$0\underbrace{1\ldots1}_{m_1+1}0\underbrace{1\ldots1}_{m_2+1}0\,,$$
$$\uparrow$$
$$q_1$$

то в заключительной конфигурации получим код числа $m_1 + m_2 + 2$. То есть машина M_3 вычисляет также функцию $f(x, y) = x + y + 2$.

Удвоение слова. Пусть $A = \{a_0, a_1, \ldots, a_k\}$, $a_0 = 0$. Построим машину Тьюринга M^D, которая удваивает слова в алфавите A и помещает между двумя экземплярами слова символ $\square \notin A$: $a_{i_1}a_{i_2}\ldots a_{i_r} \to a_{i_1}\ldots a_{i_r}\square\, a_{i_1}\ldots a_{i_r}$, где $a_{i_1}, \ldots, a_{i_r} \neq 0$, $r \geqslant 1$. То есть эта машина переводит любую конфигурацию вида

$$a_{i_1}a_{i_2}\ldots a_{i_r},$$
$$\uparrow$$
$$q_1$$

$a_{i_1}, \ldots, a_{i_r} \neq 0$, в конфигурацию вида

$$a_{i_1}a_{i_2}\ldots a_{i_r}\square\, a_{i_1}a_{i_2}\ldots a_{i_r}.$$
$$\uparrow$$
$$q_0$$

Машина M^D работает в расширенном алфавите

$$A' = \{0, a_1, \ldots, a_k, \square, a'_1, \ldots, a'_k\},$$

где $a'_1, \ldots, a'_k$ — "двойники" букв $a_1, \ldots, a_k$ соответственно, $a'_1, \ldots, a'_k \notin A$. Работа машины M^D состоит из нескольких циклов. В очередном цикле головка отмечает место, где стоит очередная буква, запоминает ее и переносит эту букву через необработанную часть слова, символ $\square$ и через построенную к этому моменту часть второго экземпляра слова, ставит букву на первое пустое место и идет за следующей буквой. После того как все буквы перенесены, отметки стираются и головка возвращается в исходную позицию.

Машина M^D имеет множество состояний

$$\{q_0,\ q_1,\ q_4,\ q_5,\ q_6,\ q_{2,i},\ q_{3,i},\ i = 1, \ldots, k\};$$

программа машины M^D имеет следующий вид:

$q_1 a_i$	$\to$	$q_{2,i}a'_i R$	запоминание буквы a_i;
$q_{2,i}a_j$	$\to$	$q_{2,i}a_j R$	перенос буквы a_i через необработанную часть слова;
$q_{2,i}0$	$\to$	$q_{3,i}\square R$	запись символа $\square$;
$q_{2,i}\square$	$\to$	$q_{3,i}\square R$	перенос a_i через символ $\square$;
$q_{3,i}a_j$	$\to$	$q_{3,i}a_j R$	перенос a_i через построенную часть второго экземпляра слова;
$q_{3,i}0$	$\to$	$q_4 a_i L$	запись a_i на первом пустом месте;
$q_4 a_i$	$\to$	$q_4 a_i L$	движение влево через построенную часть второго экземпляра слова;
$q_4\square$	$\to$	$q_5\square L$	переход через символ $\square$;
$q_5 a_i$	$\to$	$q_5 a_i L$	движение влево через необработанную часть слова;
$q_5 a'_i$	$\to$	$q_1 a'_i R$	обнаружение левой необработанной буквы;
$q_1\square$	$\to$	$q_6\square L$	обнаружение, что все буквы перенесены;
$q_6 a'_i$	$\to$	$q_6 a_i L$	стирание штрихов;
$q_6 0$	$\to$	$q_0 0 R$	обнаружение, что все штрихи стерты,

где $i, j = 1, 2, \ldots, k$.

Несложно убедиться, что построенная машина является искомой.

Выдвигается следующее предположение.

Тезис Тьюринга. *Любой интуитивно осуществимый алгоритм может быть реализован в некоторой машине Тьюринга.*

Тезис Тьюринга не является теоремой, его нельзя доказать из-за того, что понятие интуитивно осуществимого алгоритма является неформальным. Можно лишь приводить доводы в его пользу. Прежде всего это предположение подтверждается опытом людей, занимавшихся построением алгоритмов в течение нескольких тысяч лет. Важным доводом в пользу этого тезиса является также следующий. Как уже говорилось, рядом авторов были предложены разные формальные уточнения понятия алгоритма. Все они оказались эквивалентными.

Теперь мы можем указать некоторые проблемы, являющиеся алгоритмически неразрешимыми.

Покажем, что все машины Тьюринга можно закодировать словами в алфавите $\{1, *\}$ так, чтобы по программе машины эффективно строился код и, наоборот, по коду эффективно восстанавливалась программа.

Пусть M — произвольная машина Тьюринга с входным алфавитом $A = \{a_0, a_1, \ldots, a_k\}$ и множеством состояний $Q = \{q_0, q_1, \ldots, q_n\}$. Занумеруем все символы, встречающиеся в командах машины M, целыми неотрицательными числами:

$$\begin{array}{cccccccccccc} S & L & R & a_0 & a_1 & \ldots & a_k & q_0 & q_1 & \ldots & q_n \\ 0 & 1 & 2 & 3 & 4 & \ldots & k+3 & k+4 & k+5 & \ldots & k+n+4\,. \end{array}$$

В результате каждому символу b из множества

$$\{S,\, L,\, R,\, a_0,\, a_1, \ldots, a_k,\, q_0,\, q_1, \ldots, q_n\}$$

сопоставлено некоторое число $n(b) \in N$. Как и ранее, число $n(b)$ будем задавать словом из единиц длины $n(b)+1$; будем это слово называть *кодом* символа b и обозначать через $K(b)$. Команде F вида $qa \to q'a'd$ машины M сопоставим слово

$$K(q) * K(a) * K(q') * K(a') * K(d)$$

(*код команды*); обозначим код команды F через $K(F)$. Пусть $F_1, \ldots, F_l$ — список всех команд машины M. Сопоставим машине M слово $K(F_1) * K(F_2) * \cdots * K(F_l)$. Это слово будем называть *кодом машины* M и обозначать через $K(M)$. Из построения видно, что он эффективно определяется системой команд. Легко видеть также, что по коду $K(M)$ машины эффективно определяется система команд. В самом деле, просматривая в кодах команд вторые компоненты, можно найти число k. После этого по коду $K(q)$ состояния q легко определяется номер этого состояния.

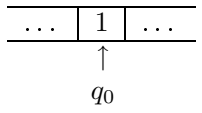
Далее будем рассматривать машины, входные алфавиты которых содержат символы $0, 1$ и $*$.

Машина M, применимая к слову $K(M)$ (т. е. к собственному коду), называется *самоприменимой*. Машина, неприменимая к собственному коду, называется *несамоприменимой*.

Каждая машина является либо самоприменимой, либо несамоприменимой. Существуют самоприменимые машины. Примерами таких машин являются машины, в правых частях команд которых встречается только заключительное состояние, в частности машина M_2 из первого примера. Существуют и несамоприменимые машины. Примерами таких машин являются машины, в правых частях команд которых не встречается заключительное состояние; к их числу относится машина M_1 (из первого примера).

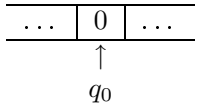
Проблема самоприменимости. *Проблема самоприменимости* состоит в следующем: по коду $K(M)$ произвольной машины M установить, является она самоприменимой или нет. Или, точнее, найти алгоритм для решения этой задачи. Поскольку мы приняли тезис Тьюринга, алгоритм должен быть выражен в виде соответствующей машины Тьюринга. Таким образом, требуется построить машину Тьюринга, которая была бы применима к кодам всех машин и заключительные конфигурации при работе над кодами самоприменимых и несамоприменимых машин эффективно бы различались. Будем говорить, что машина M^S решает проблему самоприменимости, если

1) M^S применима к коду любой машины;
2) в случае самоприменимой машины заключительная конфигурация имеет вид



;

3) в случае несамоприменимой машины заключительная конфигурация имеет вид



(вне обозреваемой головкой ячейки может стоять что угодно).

Теорема 1. *Проблема самоприменимости алгоритмически неразрешима. т. е. не существует машины Тьюринга, решающей эту проблему в указанном выше смысле.*

Доказательство. Допустим, что существует некоторая машина M^S, решающая проблему самоприменимости. Построим новую машину $\widehat{M}$, которая применима к кодам несамоприменимых машин и неприменима к кодам самоприменимых машин.

Машина $\widehat{M}$ получается из машины M^S следующим образом. Добавим к множеству Q состояний машины M_S новое состояние $q_0' \notin Q$, которое будем считать заключительным состоянием машины $\widehat{M}$; при этом заключительное состояние q_0 машины M^S будет незаключительным ("обычным") состоянием машины $\widehat{M}$. Сохраним все команды машины M^S и добавим к ним две новые команды

$$q_0 1 \to q_0 1 S, \quad q_0 0 \to q_0' 0 S.$$

В результате получим машину $\widehat{M}$ (рис. 3).

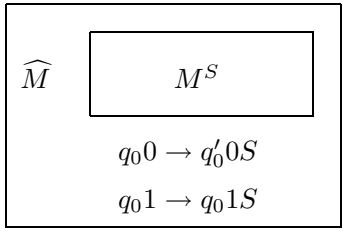


Рис. 3

Легко видеть, что она является искомой. То есть если существует машина M^S, то существует и машина $\widehat{M}$, обладающая указанными выше свойствами.

Рассмотрим функционирование машины $\widehat{M}$ над собственным кодом $K(\widehat{M})$. Машина $\widehat{M}$ является либо самоприменимой, либо несамоприменимой.

В первом случае она применима к собственному коду $K(\widehat{M})$, т. е. к коду самоприменимой машины, а это невозможно, поскольку по построению она неприменима к кодам самоприменимых машин. Во втором случае она неприменима к коду $K(\widehat{M})$, т. е. к коду несамоприменимой машин, а это невозможно, поскольку $\widehat{M}$ применима к кодам несамоприменимых машин. Таким образом, мы пришли к противоречию. Следовательно, такой машины $\widehat{M}$ не существует. Поэтому не существует и машины M^S, решающей проблему самоприменимости.

Пусть M_1 и M_2 — две машины с одинаковым входным алфавитом $A = \{a_0, a_1, \ldots, a_k\}$, имеющие множества состояний $\{q_0, q_1, \ldots, q_n\}$ и $\{q_0, q_1', \ldots, q_r'\}$ соответственно. Построим третью машину M с тем же входным алфавитом A и множеством состояний $\{q_0, q_1, \ldots, q_n, q_{n+1}, \ldots, q_{n+r}\}$, работа которой равносильна последовательной работе машин M_1 и M_2. А именно сначала исходное слово на ленте перерабатывается машиной M_1, затем если в некоторый момент времени машина M_1 переходит в заключительное состояние q_0, то в следующий момент времени содержимое ленты начинает перерабатывать машина M_2, головка которой находится в том же месте ленты, где остановилась головка машины M_1. Машина M называется *произведением* машины M_1 и машины M_2. Программа машины M получается следующим образом. Во всех командах машины M_1, содержащих символ заключительного состояния q_0 машины M_1, заменим q_0 на символ q_{n+1}; все остальные символы в программе M_1 оставим без изменения. Во всех командах машины M_2 символ q_0 (заключительное состояние машины M_2) оставим без изменения, а каждый символ q_i' заменим на символ q_{n+i}, $i = 1, \ldots, r$. Совокупность всех команд машин M_1 и M_2, измененных указанным выше способом, и будет программой машины M. Начальным состоянием машины M будет начальное состояние q_1 машины M_1, а заключительным — заключительное состояние q_0 машины M_2.

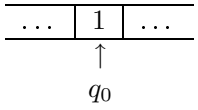
Произведение машины M_1 и машины M_2 обозначается через $M_1 \cdot M_2$.

Проблема применимости. Пусть A — произвольное конечное множество, содержащее символы $0, 1, *$ и не содержащее символ

$\square$, M — произвольная машина Тьюринга с входным алфавитом A, а $\mathfrak{A}$ — конечное слово в алфавите $A \setminus \{0\}$.

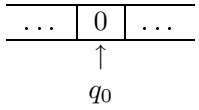
Рассмотрим следующую задачу: по коду $K(M)$ машины M и слову $\mathfrak{A}$ узнать, применима M к слову $\mathfrak{A}$ или нет. Эта задача называется *проблемой применимости*. В терминах машин Тьюринга она формулируется следующим образом. Будем говорить, что машина M^P решает проблему применимости (к словам в алфавите $A \setminus \{0\}$), если:

1) M^P применима к любому слову вида $K(M) \square \mathfrak{A}$, где $K(M)$ — код некоторой машины с входным алфавитом A, а $\mathfrak{A}$ — некоторое конечное слово в алфавите $A \setminus \{0\}$;
2) в случае, когда M применима к $\mathfrak{A}$, заключительная конфигурация машины M^P имеет вид



;

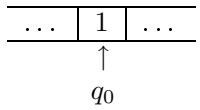
3) в случае, когда M неприменима к $\mathfrak{A}$, заключительная конфигурация имеет вид



(вне обозреваемой головкой ячейки может стоять что угодно).

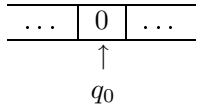
Теорема 2. *Не существует машины Тьюринга, решающей проблему применимости.*

Доказательство. Пусть A — некоторый конечный алфавит, такой, что $0, 1, * \in A$, $\square \notin A$. Предположим, что существует машина M^P, решающая проблему применимости к словам в алфавите $A \setminus \{0\}$. Пусть M^D — машина Тьюринга, удваивающая слова в алфавите $\{1, *\}$. Построим машину $\widehat{M} = M^D \cdot M^P$ — произведение машины M^D и машины M^P, работа которой заключается в следующем: сначала работает машина M^D, затем машина M^P. Легко видеть, что при работе над кодом $K(M)$ произвольной машины M машина $\widehat{M}$ сначала запишет на ленте слово $K(M) \square K(M)$ (причем головка будет обозревать самый левый символ этого слова), а затем заставит работать над этим словом машину M^P. Машина M^P применима ко всем таким словам и остановится в конфигурации



,

когда машина M применима к слову $K(M)$ (т. е. когда M самоприменима) и остановится в конфигурации



,

когда в свою очередь M неприменима к слову $K(M)$ (т. е. когда M несамоприменима). Таким образом, машина $\widehat{M}$ решает проблему самоприменимости. В силу теоремы 1 это невозможно. Поэтому машины M^P не существует.

Отметим, что в этой теореме доказано большее: алгоритмическая неразрешимость проблемы применимости к словам в алфавите $\{1, *\}$.

Лекция № 12

Пусть A — конечный алфавит, содержащий символы $0, 1, *$ и не содержащий символов α и $\square$. Пусть M — машина Тьюринга с входным алфавитом A и множеством состояний Q. Будем кодировать конфигурации машины M, в которых на ленте записано конечное число непустых символов, словами в алфавите $A \cup \{\alpha\}$.

Пусть конфигурация T машины M имеет вид

$a^{(1)}$	$\ldots$	$a^{(i-1)}$	$a^{(i)}$	$a^{(i+1)}$	$\ldots$	$a^{(m)}$
			$\uparrow$			
			q			

,

где $a^{(1)}, \ldots, a^{(m)} \in A$, $q \in Q$, $a^{(1)}, a^{(m)} \neq 0$, и все непустые символы, записанные на ленте, находятся среди букв $a^{(1)}, a^{(2)}, \ldots, a^{(m)}$. Сопоставим конфигурации T слово

$$a^{(1)}\, a^{(2)} \ldots \alpha\, K(q)\, \alpha\, a^{(i)} \ldots a^{(m)}$$

(код конфигурации); обозначим код конфигурации T через $N(T)$. Конфигурацию T будем называть *правильной*, если среди $a^{(1)}, a^{(2)}, \ldots, a^{(m)}$ нет пустых символов.

Слова, записанные на ленте в правильных конфигурациях, будем называть *правильными*; к правильным словам отнесем также слово, состоящее только из пустых символов.

Проблема переводимости. Рассмотрим следующую задачу: по коду $K(M)$ машины M и кодам $N(T_1)$ и $N(T_2)$ конфигураций T_1 и T_2 соответственно узнать, переводит машина M конфигурацию T_1 в конфигурацию T_2 или нет. Эта задача называется *проблемой переводимости* (конфигураций). В терминах машины Тьюринга она формулируется следующим образом. Будем говорить, что машина M^K решает проблему переводимости (конфигураций), если:

1) машина M^K применима к любым словам вида

$$K(M) \square N(T_1) \square N(T_2),$$

где $K(M)$ — код некоторой машины Тьюринга M, а $N(T_1)$ и $N(T_2)$ — коды некоторых конфигураций T_1 и T_2 соответственно;

2) в случае, когда машина M переводит конфигурацию T_1 в конфигурацию T_2, заключительная конфигурация машины M^K имеет вид

$\ldots$	0	1	0	$\ldots$
		$\uparrow$		
		q_0		

;

3) в случае, когда машина M не переводит конфигурацию T_1 в конфигурацию T_2, заключительная конфигурация машины M^K имеет вид

$\ldots$	0	0	0	$\ldots$
		$\uparrow$		
		q_0		

(вне обозреваемой головкой ячейки стоят пустые символы).

Следует отметить, что приведенное выше кодирование машины M и конфигураций T_1 и T_2 в начальной конфигурации T машины M^K не является существенным. Важно лишь, чтобы по T эффективно определялись машина M и конфигурации T_1 и T_2 и, наоборот, по машине M и конфигурациям T_1 и T_2 эффективно строилась конфигурация T.

Теорема 1. *Не существует машины Тьюринга, решающей проблему переводимости.*

Докажем сначала следующую лемму.

Лемма. *Для любой машины Тьюринга M существует машина Тьюринга $\widehat{M}$, обладающая следующими свойствами:*

1) *если начальная конфигурация T_0 правильная и M применима к T_0, то $\widehat{M}$ также применима к T_0;*
2) *если M неприменима к конфигурации T_0, то $\widehat{M}$ также неприменима к T_0;*
3) *заключительная конфигурация машины $\widehat{M}$ имеет вид*

$\ldots$	0	$*$	0	$\ldots$
		$\uparrow$		
		q_0		

(*вне обозреваемой головкой ячейки стоят пустые символы, q_0 — заключительное состояние машины $\widehat{M}$*).

Доказательство. Пусть M — произвольная машина Тьюринга с входным алфавитом $A = \{a_0, a_1, \ldots, a_k\}$, где $a_0 = 0$, и множеством состояний $Q = \{q_0, q_1, \ldots, q_n\}$, где q_0 — заключительное состояние машины.

Построим сначала машину M_1, которая удовлетворяет свойствам 1 и 2 из условия леммы, а также свойству

3′) если начальная конфигурация T_0 машины M_1 правильная, то все последующие конфигурации машины M_1 также правильные.

Пусть символ (непустой) 0_1 не содержится в A. Машина M_1 имеет входной алфавит $A_1 = A \cup \{0_1\}$ и множество состояний $Q_1 = \{q_0, q_1, \ldots, q_n, q_{n+1}\}$. Символ 0_1 она воспринимает как 0 и пишет вместо 0. Начальное состояние q_0 машины M становится обычным состоянием q_{n+1} машины M_1; заключительным состоянием машины M_1 является q_0. Программа машины M_1 строится следующим образом.

Сначала каждая команда $F = qa \to q'a'd$, где $q \in Q \setminus \{q_0\}$, $a \in A$, преобразуется в соответствии со следующими правилами:

a) если $a \neq 0$ и $a' \neq 0$, то команда F остается без изменения;

b) если $a \neq 0$, $a' = 0$, то команда $F = qa \to q'0d$ заменяется на команду $F' = qa \to q'0_1d$;

c) если $a = 0$, $a' \neq 0$, то команда $F = q0 \to q'a'd$ остается без изменения и, кроме того, к списку команд добавляется команда $F' = q0_1 \to q'a'd$;

d) если $a = a' = 0$, то команда $F = q0 \to q'0d$ заменяется на две команды $F' = q0 \to q'0_1d$ и $F'' = q0_1 \to q'0_1d$.

Затем во всех полученных командах, содержащих символ q_0 заключительного состояния машины M, символ q_0 заменяется на символ q_{n+1}; все остальные символы остаются без изменения.

Кроме того, образуем $k + 2$ новые команды

$$q_{n+1}a \to q_0aS,$$

где $a \in A_1$, и добавим их к полученному ранее списку команд.

В результате получим программу машины M_1. Легко видеть, что M_1 удовлетворяет свойствам 1, 2 и 3′.

Рассмотрим теперь машину M_2 с входным алфавитом $A \cup \{0_1\}$,

множеством состояний $Q_2 = \{q_0, q_1', q_2'\}$ (q_1' — начальное, q_0 — заключительное состояния) и программой

$$\begin{aligned} q_1'a &\to q_1'aR \quad (a \neq 0),\\ q_1'0 &\to q_2'0L,\\ q_2'a &\to q_2'0L \quad (a \neq 0),\\ q_2'0 &\to q_0 * S. \end{aligned}$$

Машина M_2 применима к любому правильному слову; при этом заключительная конфигурация машины M_2 имеет вид

…	0	*	0	…
		↑		
		q_0		

.

Построим машину $\widehat{M} = M_1 \cdot M_2$ — произведение машин M_1 и M_2. Легко видеть, что машина $\widehat{M}$ удовлетворяет условиям леммы.

Доказательство теоремы 1. Предположим, что существует машина M^K, решающая проблему переводимости конфигураций. Покажем тогда, что существует машина R, решающая проблему применимости для правильных слов.

Машина R состоит из четырех частей — машин R_1, R_2, R_3 и M^K. Опишем работу каждой из этих машин, указывая при этом слова, записанные на ленте в заключительных конфигурациях.

Машина R_1 начинает работать над словом

$$K(M) \square \mathfrak{A},$$

где $K(M)$ — код некоторой машины Тьюринга, а $\mathfrak{A}$ — произвольное правильное слово в алфавите A. Машина R_1 преобразует код $K(M)$ в код $K(\widehat{M})$ машины $\widehat{M}$, соответствующей машине M (см. лемму), и строится в соответствии с эффективной процедурой, указанной при доказательстве леммы. В результате работы машины R_1 получается слово

$$K(\widehat{M}) \square \mathfrak{A}.$$

Машина R_2 преобразует слово $\mathfrak{A} = a^{(1)} \ldots a^{(m)}$ над алфавитом $A \setminus \{0\}$ в код $N(T_0)$ начальной конфигурации T_0 машины $\widehat{M}$:

$a^{(1)}$	…	$a^{(m)}$
↑		
q_1		

,

т. е. в слово $N(T_0) = \alpha\, K(q_1)\, \alpha a^{(1)} \dots a^{(m)}$. Для этого по коду $K(\widehat{M})$ машина R_2 сначала определяет число букв в алфавите A, а затем строит код $K(q_1)$ начального состояния q_1 машины $\widehat{M}$. В результате работы машины R_2 получается слово

$$K(\widehat{M}) \,\square\, N(T_0).$$

Машина R_3 сначала дописывает справа символ $\square$, затем строит код $N(T_1)$ заключительной конфигурации T_1 машины $\widehat{M}$, которая имеет вид

…	0	*	0	…
		↑		
		q_0		

,

т. е. слово

$$N(T_1) = \alpha\, K(q_0)\, \alpha *$$

(отметим, что все машины, построенные в соответствии с леммой, имеют заключительную конфигурацию такого вида), и приписывает его справа к полученному ранее слову. В результате работы машины R_3 получается слово

$$K(\widehat{M}) \,\square\, N(T_0) \,\square\, N(T_1).$$

Наконец, начинает работу машина M^K. Она применима ко всем словам такого типа и приводит к заключительной конфигурации

…	0	1	0	…
		↑		
		q_0		

,

когда машина $\widehat{M}$ переводит конфигурацию T_0 в конфигурацию T_1 (т. е. когда M применима к слову A) и приводит к заключительной конфигурации

…	0	0	0	…
		↑		
		q_0		

,

когда машина $\widehat{M}$ не переводит конфигурацию T_0 в конфигурацию T_1 (т. е. когда M неприменима к слову A).

Таким образом, построенная машина R решает проблему применимости (для правильных слов). В силу доказанной ранее теоремы и замечания к ней это невозможно. Поэтому машины M^K не существует.

Отметим, что в данной теореме доказано большее: алгоритмическая неразрешимость переводимости в заключительное состояние.

Проблема эквивалентности слов в ассоциативных исчислениях. Мы привели несколько примеров алгоритмически неразрешимых задач. Все эти задачи были связаны с работой машин Тьюринга. Укажем теперь пример внешней алгоритмически неразрешимой задачи — задачи эквивалентности слов. Введем понятие ассоциативного исчисления.

Пусть A — конечный алфавит. *Ассоциативное исчисление* U задается конечным набором допустимых подстановок, т. е. пар слов в алфавите A:

$$P_1 \leftrightarrow Q_1,$$
$$P_2 \leftrightarrow Q_2,$$
$$\dots$$
$$P_l \leftrightarrow Q_l.$$

Считаем, что множеству A^* всех конечных слов в алфавите A принадлежит также пустое слово Λ, не содержащее символов. Пустое слово Λ обладает следующим свойством: для любого слова α в алфавите A выполняются равенства $\Lambda\alpha = \alpha\Lambda = \alpha$.

Подстановка $P_i \leftrightarrow Q_i$ $(1 \leqslant i \leqslant l)$ задает следующее элементарное преобразование слов в алфавите A: любое слово вида $\alpha P_i \beta$, где $\alpha, \beta \in A^*$, можно преобразовать в слово $\alpha Q_i \beta$ (применение подстановки $P_i \leftrightarrow Q_i$ слева направо) и, наоборот, из слова $\alpha Q_i \beta$ можно получить слово $\alpha P_i \beta$ (применение подстановки $P_i \leftrightarrow Q_i$ справа налево). Два слова R и T называются *эквивалентными* в ассоциативном исчислении U, если существует последовательность слов $\alpha_1, \alpha_2, \dots, \alpha_m$ в алфавите A, такая, что $R = \alpha_1$, $T = \alpha_m$ и каждое слово α_{i+1} $(1 \leqslant i < m)$ получается из слова α_i применением некоторой допустимой подстановки (слева направо или справа налево).

Пример. Пусть $A = \{a, b, c\}$, а система допустимых подстановок имеет вид

$$a \leftrightarrow aa,$$
$$b \leftrightarrow bc,$$
$$ba \leftrightarrow bc.$$

Тогда, например, слово $aaabca$ эквивалентно слову aab в рассматриваемом исчислении:

$$a\underline{aa}bca,\ aa\underline{bc}a,\ aa\underline{ba},\ aa\underline{bc},\ aab.$$

Возникает следующая *проблема эквивалентности*: по произвольному ассоциативному исчислению и по двум произвольным словам этого исчисления узнать, эквивалентны эти слова в данном исчислении или нет.

Теорема 2. *Проблема эквивалентности для ассоциативных исчислений алгоритмически неразрешима.*

Доказательство. Утверждение теоремы следует из алгоритмической неразрешимости проблемы переводимости конфигураций в машинах Тьюринга.

Ячейка ленты называется активной, если либо в ней записан непустой символ, либо ее обозревает головка, либо она находится между двумя ячейками, одну из которых обозревает головка, а в другой записан непустой символ.

Сопоставим каждой машине Тьюринга M некоторое ассоциативное исчисление U_M.

Пусть M имеет входной алфавит $A = \{a_0, a_1, \ldots, a_k\}$, $a_0 = 0$, и множество состояний $Q = \{q_0, q_1, \ldots, q_n\}$, q_0 — заключительное состояние. Программа машины M для всех q_i, a_j (где $i = 1, \ldots, n$; $j = 0, 1, \ldots, k$) содержит в точности одну команду, которая начинается словом $q_i a_j$.

Исчисление U_M имеет алфавит $A_M = A \cup Q \cup \{\lambda, \pi\}$, где символы λ, π не принадлежат $A \cup Q$.

Рассмотрим конфигурацию T машины M

$a^{(1)}$	$\ldots$	$a^{(i-1)}$	$a^{(i)}$	$a^{(i+1)}$	$\ldots$	$a^{(m)}$
			$\uparrow$			
			q			

,

в которой все символы $a^{(1)}, \ldots, a^{(m)}$ находятся в активных клетках. Сопоставим конфигурации T слово

$$\lambda\, a^{(1)} \ldots a^{(i-1)}\, q\, a^{(i)}\, a^{(i+1)} \ldots a^{(m)}\, \pi$$

(K-слово конфигурации T). Обозначим K-слово конфигурации T через $K(T)$; K-слова заключительных конфигураций называются заключительными K-словами.

Сопоставим командам машины M следующие допустимые подстановки:

a) команде $qa \to q'a'S$ сопоставляются подстановки

$$\begin{aligned} bqac &\leftrightarrow bq'a'c, \\ \lambda qac &\leftrightarrow \lambda q'a'c, \\ bqa\pi &\leftrightarrow bq'a'\pi, \\ \lambda qa\pi &\leftrightarrow \lambda q'a'\pi\ , \end{aligned}$$

где b и c — любые символы из A;

b) команде $qa \to q'a'R$ сопоставляются подстановки

$$\begin{aligned} bqac &\leftrightarrow ba'q'c, \\ \lambda qac &\leftrightarrow \lambda a'q'c \quad (\text{если } a' \neq 0), \\ \lambda qac &\leftrightarrow \lambda q'c \quad (\text{если } a' = 0), \\ bqa\pi &\leftrightarrow ba'q'0\pi, \\ \lambda qa\pi &\leftrightarrow \lambda a'q'0\pi \quad (\text{если } a' \neq 0), \\ \lambda qa\pi &\leftrightarrow \lambda q'0\pi \quad (\text{если } a' = 0), \end{aligned}$$

где b и c — любые символы из A;

c) команде $qa \to q'a'L$ сопоставляются подстановки

$$\begin{aligned} bqac &\leftrightarrow qba'c, \\ \lambda qac &\leftrightarrow \lambda q'0a'c, \\ bqa\pi &\leftrightarrow q'ba'\pi \quad (\text{если } a' \neq 0), \\ bqa\pi &\leftrightarrow q'b\pi \quad (\text{если } a' = 0), \\ \lambda qa\pi &\leftrightarrow \lambda q'0a'\pi \quad (\text{если } a' \neq 0), \\ \lambda qa\pi &\leftrightarrow \lambda q'0\pi \quad (\text{если } a' = 0), \end{aligned}$$

где b и c — любые символы из A.

Нетрудно показать, что допустимые подстановки исчисления U_M обладают следующими свойствами:

1) к каждому незаключительному K-слову применима слева направо не более чем одна подстановка, приводящая не более чем к одному слову;

2) к каждому заключительному K-слову не применима слева направо никакая подстановка;

3) если $K(T_1)$ — K-слово конфигурации T_1 и к этому слову слева применима слева направо некоторая подстановка, в результате

чего получается слово K_2, то K_2 есть K-слово некоторой конфигурации T_2, в которую машина M переведет конфигурацию T_1 за один шаг.

Докажем теперь следующее утверждение.

Пусть $K_1 = K(T_1)$ — K-слово произвольной конфигурации T_1, $K_2 = K(T_2)$ — K-слово заключительной конфигурации T_2. K-слово K_1 эквивалентно K-слову K_2 в исчислении U_M тогда и только тогда, когда машина M переводит конфигурацию T_1 в конфигурацию T_2.

В самом деле, если машина M переводит конфигурацию T_1 в конфигурацию T_2, то в силу свойства 3 K-слово K_1 эквивалентно K-слову K_2 в исчислении U_M.

Покажем справедливость этого утверждения в другую сторону. Пусть слова K_1 и K_2 эквивалентны в исчислении U_M. Тогда существует последовательность

$$\alpha_1, \alpha_2, \ldots, \alpha_t$$

слов в алфавите A_M, такая, что $\alpha_1 = K_1$, $\alpha_t = K_2$ и α_{j+1} $(1 \leqslant j < t)$ получается из α_j применением некоторой подстановки исчисления U_M (слева направо или справа налево).

Поместим между α_j и α_{j+1} $(j = 1, \ldots, t-1)$ символ $\rightarrow$, если слово α_{j+1} получено из слова α_j применением подстановки слева направо, и символ $\leftarrow$, если подстановка применялась справа налево. В результате получим последовательность I

$$\alpha_1\, e_1\, \alpha_2\, e_2 \ldots e_{t-1}\, \alpha_t,$$

где $e_1, \ldots, e_{t-1} \in \{\rightarrow, \leftarrow\}$. Образуем теперь новую последовательность, которая также начинается словом α_1 и заканчивается словом α_t и в которой все стрелки направлены слева направо.

Заметим, что в силу свойства 2 исчисления U_M выполняется равенство $e_{t-1} = \rightarrow$. Если в последовательности I все стрелки направлены слева направо, то она является искомой. В противном случае найдем в последовательности I самую правую стрелку, направленную справа налево. Пусть, например, $e_{i-1} = \leftarrow$, $1 < i < t$. Соответствующий фрагмент последовательности имеет вид

$$\alpha_{i-1} \leftarrow \alpha_i \rightarrow \alpha_{i+1}.$$

Из этого следует, что слова α_{i-1} и α_{i+1} получены из слова α_i применением некоторых подстановок исчисления U_M слева направо.

Поэтому в силу свойства 1 $\alpha_{i-1} = \alpha_{i+1}$. Следовательно, из последовательности I можно удалить фрагмент $\alpha_{i-1} \leftarrow \alpha_i \rightarrow$. В результате получится более короткая последовательность I_1

$$\alpha_1\, e_1\, \alpha_2 \ldots e_{i-2}\, \alpha_{i+1} \rightarrow \alpha_{i+2} \ldots \rightarrow \alpha_t.$$

Продолжая этот процесс, на некотором шаге получим последовательность, в которой все стрелки направлены слева направо. Этой последовательности будет соответствовать последовательность конфигураций, которая начинается конфигурацией T_1, заканчивается конфигурацией T_2, причем каждая следующая конфигурация получается из предыдущей в результате выполнения одной команды машины M. Тем самым утверждение доказано.

Таким образом, поскольку проблема переводимости в заключительную конфигурацию алгоритмически неразрешима, то в силу доказанного выше утверждения алгоритмически неразрешима и проблема эквивалентности произвольного K-слова заключительному K-слову в исчислении U_M, а поэтому и общая проблема эквивалентности.

Теорема полностью доказана.

Лекция № 13

Нормальные алгорифмы (А. А. Маркова). Рассмотрим некоторый конечный алфавит $A = \{a_1, \ldots, a_k\}$. Обозначим через A^* множество всех конечных слов в алфавите A. Будем предполагать, что A^* содержит также пустое слово, в котором нет символов; пустое слово обозначается через Λ. Для любых двух слов $\alpha = a_{i_1}a_{i_2}\ldots a_{i_k}$ и $\beta = b_{j_1}b_{j_2}\ldots b_{j_n}$ через $\alpha\beta$ обозначается слово $a_{i_1}a_{i_2}\ldots a_{i_k}b_{j_1}b_{j_2}\ldots b_{j_n}$. Для пустого слова Λ и любого слова $\alpha \in A^*$ выполняются равенства $\Lambda\alpha = \alpha\Lambda = \alpha$. Слово β называется подсловом слова α, если найдутся слова γ и δ из A^*, такие, что $\alpha = \gamma\beta\delta$; в этом случае говорят также, что слово β входит в α. Очевидно, что пустое слово Λ является подсловом любого слова из A^*.

Будем предполагать, что алфавит A не содержит символов $\to$ и $\cdot$ в качестве букв. Подстановками в алфавите A называются выражения вида

$$P \to Q$$

(незаключительная подстановка) или

$$P \to \cdot Q$$

(заключительная подстановка), где P и Q — слова из A^*. Слова P и Q называются соответственно левой и правой частью подстановки.

Пусть F — некоторая подстановка в алфавите A, а α — некоторое слово из A^*. Постановка F *применима* к α, если левая часть подстановки входит в α. В противном случае F *неприменима* к α. В том случае, когда F применима к α, *результатом* подстановки считается слово, которое получается из α заменой первого (самого левого) вхождения левой части подстановки F в слово α на правую часть подстановки; результат подстановки обозначается через $F(\alpha)$. В частности, когда левая часть подстановки есть Λ, а правая часть — некоторое слово Q, то по определению полагаем, что результатом указанной подстановки является слово $Q\alpha$.

Нормальный алгорифм в алфавите A задается конечной упорядоченной системой Σ подстановок $F_1, \ldots, F_l$ (схемой нормального алгорифма).

Нормальный алгорифм преобразует слова из A^* в слова из A^*. Работа нормального алгорифма M над словом $\alpha \in A^*$ состоит из последовательных этапов $1, 2, \ldots$ переработки слова α, в результате которых получаются слова $\alpha_1, \alpha_2, \ldots, \alpha_j, \alpha_{j+1}, \ldots$ из A^*. На первом этапе по определению полагаем $\alpha_1 = \alpha$ и говорим, что слово α_1 получено после первого этапа переработки.

Пусть после j-го этапа переработки получено слово α_j (и алгорифм M к этому моменту еще не завершил работу), $j \geqslant 1$. Тогда алгорифм работает следующим образом.

1. Если в схеме Σ нет подстановок, применимых к слову α_j, то M завершает свою работу и *результатом* $M(\alpha)$ работы алгорифма над словом α считается слово $M(\alpha) = \alpha_j$.

2. В противном случае в схеме Σ выбирается первая по порядку подстановка F, применимая к α_j, и эта подстановка применяется к слову α_j. Результат $F(\alpha_j)$ обозначается через α_{j+1}. При этом

(a) если φ — заключительная подстановка, то алгорифм M завершает свою работу, и результатом работы M над словом α является слово $M(\alpha) = \alpha_{j+1}$;

(b) если F — незаключительная подстановка, то алгорифм M переходит к следующему этапу переработки.

Таким образом, работа алгорифма M заканчивается в двух случаях:

1) когда на очередном этапе ни одна из подстановок схемы Σ не применима;

2) после применения заключительной подстановки.

Нормальный алгорифм M называется *применимым* к слову α, если он, работая в соответствии с установленными правилами, на некотором этапе $j+1$ ($j \geqslant 1$) завершил свою работу. *Результатом* работы алгорифма M над словом α в этом случае считается слово $M(\alpha)$ ($M(\alpha) = \alpha_j$, если ни одна из подстановок не применима к α_j, и $M(\alpha) = \alpha_{j+1} = F(\alpha_j)$, если на этапе $j+1$ применялась заключительная подстановка F). В противном случае (т. е. если последовательность $\alpha_1 = \alpha, \alpha_2, \ldots$ бесконечна) алгорифм M называется *неприменимым* к слову α, и в этом случае значение $M(\alpha)$ неопределено.

При рассмотрении нормального алгорифма M в алфавите A иногда интересуются только результатами его работы над словами в алфавите B, где $B \subseteq A$. Тогда говорят, что нормальный алгорифм в алфавите A есть алгорифм над алфавитом B.

Функция $f(x)$, определенная на некотором подмножестве $D(f)$ множества A^* и принимающая значения из A^*, называется *частич-*

ной словарной функцией в алфавите A; $D(f)$ — область определения функции f.

Пусть $f(x)$ — некоторая одноместная частичная словарная функция в алфавите A, а M — некоторый нормальный алгорифм над A. Алгорифм M *вычисляет* функцию f, если для любого $\alpha \in A^*$ выполняются следующие условия:

1) если $\alpha \notin D(f)$, то M неприменим к α;
2) если $\alpha \in D(f)$, то M применим к α и $M(\alpha) = f(\alpha)$.

Частичная словарная функция $f(x)$ в алфавите A называется *нормально вычислимой* (вычислимой при помощи нормальных алгорифмов), если существует нормальный алгорифм над A, который вычисляет функцию f.

Отметим, что каждый нормальный алгорифм M в алфавите A вычисляет некоторую одноместную словарную функцию $f_M(x)$ в алфавите A, в то время как нормальные алгорифмы над A (работающие в некоторых расширениях A) этим свойством, вообще говоря, не обладают, поскольку результаты работы этих алгорифмов могут содержать символы, не принадлежащие A.

Рассмотрим несколько простых примеров.

Примеры.

1) Нормальный алгорифм в алфавите $A = \{a_1, \ldots, a_k\}$ со схемой

$$a \to ,$$

где a — произвольный символ из A, вычисляет функцию, значение которой на любом слове из A^* равно Λ;

2) нормальный алгорифм в алфавите $A = \{a_1, \ldots, a_k\}$ со схемой

$$\to$$

вычисляет нигде не определенную частичную словарную функцию в алфавите A;

3) нормальный алгорифм в алфавите $A = \{a_1, \ldots, a_k\}$ со схемой

$$\to \cdot$$

вычисляет тождественную словарную функцию $f(x) = x$ в алфавите A;

4) нормальный алгорифм в алфавите $A = \{a, b, c\}$ со схемой

$$\begin{cases} a \to b \\ ab \to \cdot c \\ c \to aba \end{cases}$$

вычисляет словарную функцию $g(x)$ в алфавите A, такую, что $g(\Lambda) = \Lambda$ и для любого слова $\alpha = a_1 \ldots a_k$ из A^*, содержащего m символов c, выполняется равенство $g(\alpha) = \underbrace{b \ldots b}_{k+2m}$.

Рассмотрим более сложный пример.

Пример. Пусть $A = \{a_1, \ldots, a_k\}$ — некоторый конечный алфавит, не содержащий символов $*, \beta, \gamma$, и пусть $B = A \cup \{*, \beta, \gamma\}$. Рассмотрим нормальный алгорифм M над A со следующей схемой:

$$\begin{aligned} ab\beta &\to b\beta a, \\ *a &\to a\beta a*, \\ \beta &\to \gamma, \\ \gamma &\to \ , \\ * &\to \cdot , \\ &\to *, \end{aligned}$$

где $a, b \in A$ (т. е. эта схема при $|A| = k$ состоит из $k^2 + k + 4$ подстановки).

Можно показать, что алгорифм M удваивает слова в алфавите A, т. е. для любого слова α из A^* выполняется равенство $M(\alpha) = \alpha\alpha$.

Пример работы алгорифма M над словом $\alpha = a_1a_2a_3$ (где $a_1, a_2, a_3 \in A$):

$$\begin{aligned} \alpha_1 &= \_a_1a_2a_3; \\ \alpha_2 &= \underline{*a_1}a_2a_3; \\ \alpha_3 &= a_1\beta a_1\underline{*a_2}a_3; \\ \alpha_4 &= a_1\beta\underline{a_1a_2\beta}a_2 * a_3; \\ \alpha_5 &= a_1\beta a_2\beta a_1a_2\underline{*a_3}; \\ \alpha_6 &= a_1\beta a_2\beta a_1\underline{a_2a_3\beta}a_3*; \\ \alpha_7 &= a_1\beta a_2\beta\underline{a_1a_3\beta}a_2a_3*; \\ \alpha_8 &= a_1\underline{\beta}a_2\beta a_3\beta a_1a_2a_3*; \\ \alpha_9 &= a_1\gamma a_2\underline{\beta}a_3\beta a_1a_2a_3*; \\ \alpha_{10} &= a_1\gamma a_2\gamma a_3\underline{\beta}a_1a_2a_3*; \\ \alpha_{11} &= a_1\underline{\gamma}a_2\gamma a_3\gamma a_1a_2a_3*; \\ \alpha_{12} &= a_1a_2\underline{\gamma}a_3\gamma a_1a_2a_3*; \\ \alpha_{13} &= a_1a_2a_3\underline{\gamma}a_1a_2a_3*; \\ \alpha_{14} &= a_1a_2a_3a_1a_2a_3\underline{*}; \\ \alpha_{15} &= a_1a_2a_3a_1a_2a_3 \end{aligned}$$

(в словах α_i отмечены подслова, к которым применяются подстановки на соответствующих этапах работы алгорифма).

Пусть $f(x_1,\dots,x_n)$ — частичная числовая функция из $P_{\mathbb{N}}$ $D(f)$ — область определения f, а M — некоторый нормальный алгорифм над алфавитом $\{0,1\}$. Алгорифм M *вычисляет* функцию $f(x_1,\dots,x_n)$, если для любого набора $\alpha=(m_1,\dots,m_n)$ из $\mathbb{N}^n$ выполняются следующие условия:

1) если $\alpha\notin D(f)$, то M неприменим к слову
$$K(\alpha)=\underbrace{1\dots1}_{m_1+1}0\underbrace{1\dots1}_{m_2+1}0\dots0\underbrace{1\dots1}_{m_n+1};$$
2) если $\alpha\in D(f)$, то M применим к α и $M(K(\alpha))=K(m)$, где $m=f(m_1,\dots,m_n)$, а $K(m)=\underbrace{1\dots1}_{m+1}$.

Частичная числовая функция называется *нормально вычислимой* (вычислимой при помощи нормальных алгорифмов), если существует нормальный алгорифм над алфавитом $\{0,1\}$, который вычисляет эту функцию.

Справедливо следующее утверждение[1)].

Теорема. *Всякая частичная числовая функция вычислима при помощи нормальных алгорифмов тогда и только тогда, когда она вычислима по Тьюрингу.*

Вариантом тезиса Чёрча, относящимся к нормальным алгорифмам, является следующий принцип (А. А. Маркова).

Принцип нормализации. *Всякий алгорифм в алфавите A эквивалентен относительно A некоторому нормальному алгорифму над A.*

Частично рекурсивные функции. Пусть $f(x_1,\dots,x_n)$ — функция, определенная на подмножестве $D(f)$ множества $\mathbb{N}^n$ и принимающая значения из множества $\mathbb{N}=\{0,1,,\dots\}$, $n\geqslant 1$; $D(f)$ называется областью определения функции f (значения f на наборах из $\mathbb{N}^n\setminus D(f)$ считаются неопределенными). Будем называть такие функции частичными числовыми функциями. Множество всех частичных числовых функций обозначим через $P_{\mathbb{N}}$. Функция $f(x_1,\dots,x_n)$ из $P_{\mathbb{N}}$ называется всюду определенной числовой функцией, если $D(f)=\mathbb{N}^n$. Функции $f(x_1,\dots,x_n)$ и $g(x_1,\dots,x_n)$ называются равными (обозначение $f(x_1,\dots,x_n)=g(x_1,\dots,x_n)$), если $D(f)=D(g)$ и для любого набора $\tilde\alpha$ из $D(f)$ выполняется равенство $f(\tilde\alpha)=g(\tilde\alpha)$.

[1)] См.: *Марков А. А., Нагорный Н. М.* Теория алгорифмов. М.: Наука, 1984.

Следующие всюду определенные числовые функции называются простейшими:
$$0(x)=0,\quad s(x)=x+1,\quad I^n_m(x_1,\dots,x_n)=x_m,$$
$1\leqslant m\leqslant n,\ n=1,2,\dots$.

Определим на множестве $P_{\mathbb{N}}$ частичных числовых функций операции суперпозиции, примитивной рекурсии и минимизации.

Пусть $g(x_1,\dots,x_m), f_1(x_1,\dots,x_n),\dots,f_m(x_1,\dots,x_n)$ — некоторые частичные числовые функции, а $D(g),\ D(f_1),\dots,D(f_m)$ — области определения функций $g,f_1,\dots,f_m$ соответственно. Будем говорить, что функция
$$F(x_1,\dots,x_n)=g(f_1(x_1,\dots,x_n),\dots,f_m(x_1,\dots,x_n))$$
получается из функций $g,f_1,\dots,f_m$ при помощи операции *суперпозиции*. Значения функции F на наборе $\tilde\alpha\in\mathbb{N}$ определяются следующим образом:

1) если $\tilde\alpha\in D(f_i)$ для всех $i=1,\dots,m$ и $(f_1(\tilde\alpha),\dots,f_m(\tilde\alpha))$ принадлежит множеству $D(g)$, то
$$F(\tilde\alpha)=g(f_1(\tilde\alpha),\dots,f_m(\tilde\alpha));$$
2) в противном случае значение $F(\tilde\alpha)$ считается неопределенным.

Очевидно, что если функции $g,f_1,\dots,f_m$ всюду определенные, то и F — всюду определенная числовая функция.

Пусть заданы некоторые функции $g(x_1,\dots,x_{n-1})$ и $h(x_1,\dots,x_{n+1})$ из $P_{\mathbb{N}}$, $n\geqslant 2$. Будем говорить, что функция $f(x_1,\dots,x_n)$ получена из функций g и h с помощью операции *примитивной рекурсии* (по переменной x_n), если для всех $x_1,\dots,x_n\in\mathbb{N}$ выполняются равенства
$$\begin{cases} f(x_1,\dots,x_{n-1},0)=g(x_1,\dots,x_{n-1}),\\ f(x_1,\dots,x_{n-1},x_n+1)=h(x_1,\dots,x_{n-1},x_n,f(x_1,\dots,x_n)).\end{cases}$$
Эти равенства называются *схемой примитивной рекурсии* (по переменной x_n).

Данное определение применяется также при $n=0$. Будем говорить, что функция $f(x)$ получена из постоянной одноместной функции, равной числу $a\in\mathbb{N}$, и функции $h(x_1,x_2)\in P_{\mathbb{N}}$, если
$$\begin{cases} f(0)=a,\\ f(x+1)=h(x,f(x)).\end{cases}$$

Нетрудно показать, что для любых частичных числовых функций $g(x_1,\dots,x_{n-1})$ и $h(x_1,\dots,x_{n+1})$ существует в точности одна частичная числовая функция $f(x_1,\dots,x_n)$, получающаяся из функций g и h с помощью операции примитивной рекурсии (по переменной x_n). При этом область определения функции f удовлетворяет следующим условиям:

1) набор $(\alpha_1,\dots,\alpha_{n-1},0)$ принадлежит $D(f)$ тогда и только тогда, когда $(\alpha_1,\dots,\alpha_{n-1})\in D(g)$ $(\alpha_1,\dots,\alpha_{n-1}\in\mathbb{N})$;
2) набор $(\alpha_1,\dots,\alpha_{n-1},\alpha_n+1)$ принадлежит $D(f)$ тогда и только тогда, когда $(\alpha_1,\dots,\alpha_{n-1},\alpha_n)\in D(f)$ и одновременно $(\alpha_1,\dots,\alpha_{n-1},\alpha_n,f(\alpha_1,\dots,\alpha_n))\in D(h)$ $(\alpha_1,\dots,\alpha_n\in\mathbb{N})$.

В частности, если для некоторых $\alpha_1,\dots,\alpha_{n-1},\alpha_n$ значение $f(\alpha_1,\dots,\alpha_{n-1},\alpha_n)$ не определено, то и для всех $\beta\geqslant\alpha_n$ значения $f(\alpha_1,\dots,\alpha_{n-1},\beta)$ будут также не определены.

Из определения следует, что если функции g и h всюду определенные, то и функция f всюду определенная.

Операцию примитивной рекурсии можно применять по любой переменной.

Функция f называется *примитивно рекурсивной*, если ее можно получить конечным числом операций суперпозиции и примитивной рекурсии из простейших функций s, 0, I^n_m, $1\leqslant m\leqslant n$, $n=1,2,\dots$.

Поскольку операции суперпозиции и примитивной рекурсии, примененные ко всюду определенным функциям, дают снова всюду определенные функции, то каждая примитивно рекурсивная функция является всюду определенной числовой функцией.

Пример. Рассмотрим функцию $f(x,y)=x+y$. Она удовлетворяет соотношениям

$$\begin{cases} f(x,0)=x=I^1_1(x),\\ f(x,y+1)=(x+y)+1=s(f(x,y)).\end{cases}$$

Следовательно, функция $x+y$ получается из функций I^1_1 и $h(x,y,z)=s(I^3_3(x,y,z))=z+1$ с помощью операции примитивной рекурсии. Поскольку, согласно определению, функции I^1_1, s и I^3_3 примитивно рекурсивны, а функция h получена из функций s и I^3_3 с помощью операции суперпозиции, то функция $x+y$ примитивно рекурсивна.

Пусть $f(x_1,\dots,x_{n+1})$ — частичная числовая функция, $n\geqslant 1$. Определим функцию $g(x_1,\dots,x_n)$ следующим образом. Пусть $\tilde\alpha=(\alpha_1,\dots,\alpha_n)$ — произвольный набор из $\mathbb{N}^n$. Рассмотрим уравнение

$$f(\alpha_1,\dots,\alpha_n,y)=0:$$

a) если существует $y_0\in\mathbb{N}$, такой, что $f(\alpha_1,\dots,\alpha_n,y_0)=0$, а значения $f(\tilde\alpha,0)$, $f(\tilde\alpha,1),\dots,f(\tilde\alpha,y_0-1)$ определены и отличны от 0, то положим $g(\tilde\alpha)=y_0$;
b) в противном случае значение $g(\tilde\alpha)$ считается неопределенным.

Будем говорить, что функция $g(x_1,\dots,x_n)$ получена из функции $f(x_1,\dots,x_{n+1})$ при помощи операции *минимизации* (по переменной x_{n+1}); эту функцию обозначают следующим образом:

$$\mu_y(f(x_1,\dots,x_n,y)=0)$$

(читается как "наименьший y, такой, что $f(x_1,\dots,x_n,y)=0$"). Отметим, что значение $g(\alpha_1,\dots,\alpha_n)$ будет неопределенным в следующих случаях:

1) значение $f(\alpha_1,\dots,\alpha_n,0)$ не определено;
2) значения $f(\tilde\alpha,0),f(\tilde\alpha,1),\dots,f(\tilde\alpha,y_0-1)$ определены, но отличны от 0, а значение $f(\tilde\alpha,y_0)$ не определено;
3) значения $f(\tilde\alpha,y)$ определены для всех $y=0,1,\dots$ и отличны от 0.

Пример. Пусть $f(x,y)=x+y$. Положим $g(x)=\mu_y(f(x,y)=0)$. Очевидно, что

$$g(x)=\begin{cases}0, & \text{если } x=0;\\ \text{не определено}, & \text{если } x\neq 0.\end{cases}$$

Операцию минимизации можно также применять по любым переменным.

Частичная числовая функция называется *частично рекурсивной*, если она может быть получена из простейших функций 0, s, I^n_m, $1\leqslant m\leqslant n$, $n=1,2,\dots$, конечным числом операций суперпозиции, примитивной рекурсии и минимизации.

Всюду определенные частично рекурсивные функции называются *общерекурсивными*.

Для частичных числовых функций аналогом тезиса Тьюринга является

Тезис Чёрча. *Класс алгоритмически вычислимых функций совпадает с классом всех частично рекурсивных функций.*

Справедливо следующее утверждение[2)].

Теорема. *Множество всех частично рекурсивных функций совпадает с множеством всех частичных числовых функций, вычислимых по Тьюрингу.*

[2)] Доказательство см., например, в книге: *Мальцев А. И.* Алгоритмы и рекурсивные функции. М.: Наука, 1965.

ИСЧИСЛЕНИЕ ВЫСКАЗЫВАНИЙ

Лекция № 14

Будем рассматривать высказывания, которые бывают либо истинными, либо ложными. Например, высказывание "4 — четное число" и "$2 < 5$" являются истинными, а высказывания "4 — простое число" и "$1 + 2 = 4$" — ложными. Тем самым произвольное высказывание A можно рассматривать как величину, которая принимает значение 1, если A истинно, и 0, если A ложно. Отметим, что истинность и ложность высказываний можно обозначать и иным способом, например буквами И и Л соответственно (начальные буквы соответствующих слов). Таким образом, мы можем применять к высказываниям функции алгебры логики, получая при этом сложные высказывания. В частности, если A и B — произвольные высказывания, то

1) $A\&B$ — высказывание, истинное тогда и только тогда, когда истинны оба высказывания (в обычном языке этому высказыванию соответствует выражение "A и B");

2) $A \vee B$ — высказывание, истинное тогда и только тогда, когда хотя бы одно из высказываний A, B истинно (в обычном языке этому высказыванию соответствует выражение "A или B");

3) $\overline{A}$ — высказывание, истинное тогда и только тогда, когда высказывание A ложно (в обычном языке этому высказыванию соответствует выражение "не A");

4) $A \to B$ — высказывание, истинное тогда и только тогда, когда хотя бы одно из высказываний B, $\overline{A}$ истинно (в обычном языке этому высказыванию соответствует выражение "если A, то B"); можно считать, что соглашение об истинности этого высказывания сделано для удобства дальнейших применений;

5) $A \sim B$ — высказывание, истинное тогда и только тогда, когда высказывания A, B одновременно истинны или ложны (в обычном языке этому высказыванию соответствует выражение "A эквивалентно B").

Сложное высказывание называется *тождественно истинным*, если оно истинно при любых значениях входящих в него элементарных высказываний. Например, $A \to A$ — тождественно истинное высказывание. Аналогично сложное высказывание называется *тождественно ложным*, если оно ложно при любых значениях входящих в него элементарных высказываний. Например, $A\&\overline{A}$ — тождественно ложное высказывание. Очевидно, что сложное высказывание является тождественно истинным (соответственно тождественно ложным) тогда и только тогда, когда оно обладает тем свойством, что если подставить в него вместо элементарных высказываний некоторые буквы переменных, то полученная формула будет выражать функцию алгебры логики, тождественно равную 1 (соответственно 0).

Важным вопросом является описание множества всех формул, выражающих тождественно истинные высказывания. Один из способов такого описания заключается в следующем.

1. Указывается некоторый набор исходных формул (с этим свойством).

2. Указываются правила образования по одним формулам других.

Формулы, которые могут быть получены таким способом, и есть требуемые.

Мы ограничимся рассмотрением формул над множеством $\{^{-}, \to\}$. Хотя это множество формул и является подмножеством множества всех формул над $\{\&, \vee, ^{-}, \to, \sim, 0, 1\}$, этот язык является "не менее выразительным", поскольку все логические операции могут быть выражены через $^{-}$ и $\to$:

$$x\&y = \overline{x \to \overline{y}}, \quad x \vee y = \overline{x} \to y,$$
$$x \sim y = \overline{(x \to y) \to (\overline{y \to x})},$$
$$1 = x \to x, \quad 0 = \overline{x \to x}.$$

Поэтому формулы $A\&B$, $A \vee B$, $A \sim B$, 1, 0 над множеством $\{\&, \vee, ^{-}, \to, \sim, 0, 1\}$ могут рассматриваться как сокращенные обозначения формул

$$\frac{\overline{A \to \overline{B}}, \quad \overline{A} \to B,}{(A \to B) \to (\overline{B \to A}),}$$
$$A \to A, \quad \overline{A \to A}$$

соответственно.

Исходными формулами будут формулы следующих трех типов:

(1) $A \to (B \to A)$;

(2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;

(3) $(\overline{A} \to \overline{B}) \to (B \to A)$,

где вместо A, B и C могут подставляться любые формулы.

Формулы этих типов будем называть *аксиомами*. Сами эти выражения называются также *схемами аксиом*, так как каждое из них определяет бесконечное множество различных формул.

Введем следующее *правило вывода*:

$$\frac{A \to B, A}{B},$$

т. е. формула B получена из формул $A \to B$ и A (по правилу вывода).

Выводом называется конечная последовательность формул $F_1, F_2, \ldots, F_k$, где F_i $(1 \leqslant i \leqslant k)$ либо является аксиомой, либо получается из предыдущих формул этой последовательности по правилу вывода, т. е. существуют формулы F_j и F_r $(1 \leqslant j, r < i)$, такие, что F_j имеет вид $F_r \to F_i$. *Выводом* формулы F называется вывод, заканчивающийся формулой F. Формула F называется *выводимой*, если существует вывод этой формулы, т. е. существует вывод $F_1, \ldots, F_k$, в котором F_k совпадает с F. Выводимость формулы F обозначается следующим образом: $\vdash F$.

Аналогичным образом определяется понятие выводимой формулы из системы гипотез.

Пусть $T = \{A_1, \ldots, A_n\}$ — конечное (быть может, пустое) множество формул. Выводом из системы гипотез T называется конечная последовательность формул $F_1, F_2, \ldots, F_k$, где F_i $(1 \leqslant i \leqslant k)$ является либо аксиомой, либо одной из формул системы T, либо получается по правилу вывода из некоторых формул F_j и F_r этой последовательности с меньшими номерами $j, r < i$.

Таким образом, вывод есть вывод из пустой системы гипотез.

Пусть T — произвольное множество формул, а A, B, $A_1, \ldots, A_n$ — любые формулы. Будем писать $T, A_1, \ldots, A_n \vdash B$ вместо $T \cup \{A_1, \ldots, A_n\} \vdash B$. Отметим следующие простые свойства выводимости.

1. $T, A \vdash A$.
2. Если $T \vdash A$ и B — любая формула, то $T, B \vdash A$.
3. Если $T, B \vdash A$ и $T \vdash A$, то $T \vdash B$ (выводимую гипотезу можно удалить).
4. Если $T \vdash A_1, \ldots, T \vdash A_n$ и $A_1, A_2, \ldots, A_n \vdash B$, то $T \vdash B$ $(n \geqslant 1)$.
5. Если $T \vdash A \to B$ и $T \vdash A$, то $T \vdash B$.
6. Если $T \vdash A$ и B — любая формула, то $T \vdash B \to A$.

   В самом деле, если $F_1, F_2, \ldots, F_k$ — вывод формулы A из T, $A =_\Gamma F_k$, то[1)] выводом формулы $B \to A$ из T будет следующая последовательность формул:

$$F_1, \ldots, F_k, \quad A \to (B \to A), \quad B \to A.$$

Установим некоторые утверждения о выводимости формул.

Лемма 1. *Пусть A — произвольная формула. Тогда*

$$\vdash A \to A.$$

Доказательство. Построим вывод формулы $A \to A$.

1. $(A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$
   (подстановка в схему аксиом (2); в ней B заменено на $A \to A$, а A и C — на A).
2. $A \to ((A \to A) \to A)$
   (подстановка в схему аксиом (1); B заменено на $A \to A$).
3. $(A \to (A \to A)) \to (A \to A)$
   (из 1 и 2 по правилу вывода).
4. $A \to (A \to A)$
   (подстановка в схему аксиом (1); B заменено на A).
5. $A \to A$
   (из 3 и 4 по правилу вывода) .

Теорема 1 (о дедукции). *Пусть T — множество формул, A и B — формулы. Если $T, A \vdash B$, то $T \vdash A \to B$.*

[1)] Здесь $=_\Gamma$ — знак графического равенства.

Доказательство. Пусть $B_1, \ldots, B_n$ — вывод формулы B из $T \cup \{A\}$; $B_n =_\Gamma B$. Индукцией по n докажем, что существует вывод формулы $A \to B$ из системы T.

Пусть $n = 1$. Тогда $B_1 =_\Gamma B$. Возможны следующие случаи:

a) B — аксиома,
b) $B \in T$,
c) $B =_\Gamma A$.

По схеме аксиом (1) формула $B \to (A \to B)$ — аксиома. Поэтому в первых двух случаях $T \vdash A \to B$ по правилу вывода. В случае c формула $A \to B$ имеет вид $A \to A$. В силу леммы 1 $\vdash A \to A$. Поэтому в силу свойства выводимости 2 $T \vdash A \to A$.

Предположим, что утверждение теоремы доказано для случая, когда вывод формулы B из T, A имеет длину менее n. Докажем утверждение теоремы для случая, когда длина этого вывода равна n. Возможны следующие случаи:

a) B — аксиома,
b) $B \in T$,
c) $B =_\Gamma A$,
d) B получается по правилу вывода из некоторых формул B_i и B_k, где $i, k < n$ и B_i имеет вид $B_k \to B$.

Первые три случая рассматриваются так же, как и ранее для случая $n = 1$. Рассмотрим случай d. По индуктивному предположению имеем

1) $T \vdash A \to (B_k \to B)$,
2) $T \vdash A \to B_k$,
3) $T \vdash (A \to (B_k \to B)) \to ((A \to B_k) \to (A \to B))$
   (подстановка в схему аксиом (2); B заменено на B_k, C — на B).

По правилу вывода из 1 и 3 в силу свойства 5

4) $T \vdash (A \to B_k) \to (A \to B)$.

Наконец, по правилу вывода из 2 и 4 в силу свойства 5 окончательно получаем

5) $T \vdash A \to B$.

Отметим, что при доказательстве этой теоремы использовались только схемы аксиом (1) и (2).

Следствие. *Пусть A, B и C — формулы. Тогда*

$$A \to B,\ B \to C \vdash A \to C.$$

Доказательство. Последовательность формул

$$A \to B,\ B \to C,\ A,\ C$$

является выводом формулы C из $A \to B$, $B \to C$, A. Поэтому

$$A \to B,\ B \to C,\ A \vdash C.$$

Применяя теорему о дедукции, получаем

$$A \to B,\ B \to C \vdash A \to C.$$

Лемма 2. *Пусть A, B, C — произвольные формулы. Тогда*

a) $\vdash \overline{B} \to (B \to C)$;

b) $\vdash \overline{\overline{A}} \to A$;

c) $\vdash A \to \overline{\overline{A}}$.

Доказательство. Докажем сначала утверждение *a*.

1.	$\overline{B} \to (\overline{C} \to \overline{B})$	схема аксиом (1)
2.	$\overline{B}$	гипотеза
3.	$\overline{C} \to \overline{B}$	по правилу вывода из 1 и 2
4.	$(\overline{C} \to \overline{B}) \to (B \to C)$	схема аксиом (3)
5.	$B \to C$	по правилу вывода из 3 и 4

Таким образом, $\overline{B} \vdash B \to C$, откуда по теореме о дедукции получается соотношение *a*.

Докажем теперь утверждение *b*.

1.	$\overline{A} \to \overline{\overline{\overline{A}}}$	гипотеза
2.	$(\overline{A} \to \overline{\overline{\overline{A}}}) \to (\overline{\overline{A}} \to A)$	схема аксиом (3)
3.	$\overline{\overline{A}} \to A$	по правилу вывода из 1 и 2

В силу соотношений 1–3 $\overline{A} \to \overline{\overline{\overline{A}}} \vdash \overline{\overline{A}} \to A$; а в силу п. *a* $\overline{\overline{A}} \vdash \overline{A} \to \overline{\overline{\overline{A}}}$ (B заменено на $\overline{A}$, C — на $\overline{\overline{A}}$). Поэтому в силу свойства выводимости 4 $\overline{\overline{A}} \vdash \overline{\overline{A}} \to A$. Кроме того, $\overline{\overline{A}} \vdash \overline{\overline{A}}$. Поэтому в силу свойства (выводимости) 5 $\overline{\overline{A}} \vdash A$, откуда по теореме о дедукции получается соотношение *b*.

Установим, наконец, справедливость утверждения *c*.

1.	$(\overline{\overline{\overline{A}}} \to \overline{A}) \to (A \to \overline{\overline{A}})$	схема аксиом (3)
2.	$\overline{\overline{\overline{A}}} \to \overline{A}$	п. *b*, доказанный выше
3.	$A \to \overline{\overline{A}}$	по правилу вывода из 1 и 2

Отметим, что на самом деле в приведенном выводе вместо формулы $\overline{\overline{\overline{A}}} \to \overline{A}$ должен быть записан весь ее вывод, существующий в силу п. *b*. Мы этого не делаем для сокращения записи.

Лемма 3. *Пусть A, B и C — произвольные формулы, а T — произвольная система гипотез. Тогда*

a) $\vdash (A \to B) \to (\overline{B} \to \overline{A})$;

b) $\vdash B \to (\overline{C} \to (\overline{B \to C}))$;

c) *если* $T \vdash A \to C$ *и* $T \vdash \overline{A} \to C$, *то* $T \vdash C$.

Доказательство. Докажем сначала утверждение *a*.

В силу следствия из теоремы о дедукции

$$\overline{\overline{A}} \to A,\ A \to B,\ B \to \overline{\overline{B}} \vdash \overline{\overline{A}} \to \overline{\overline{B}}.$$

Так как в силу леммы 2 $\vdash \overline{\overline{A}} \to A$ и $\vdash B \to \overline{\overline{B}}$, то по свойству 3

$$A \to B \vdash \overline{\overline{A}} \to \overline{\overline{B}}.$$

Из схемы аксиом (3) следует, что

$$\overline{\overline{A}} \to \overline{\overline{B}} \vdash \overline{B} \to \overline{A}.$$

Поэтому в силу свойства 4 выполняется соотношение

$$A \to B \vdash \overline{B} \to \overline{A},$$

откуда п. *a* получается по теореме о дедукции.

Докажем теперь утверждение *b*.

Очевидно, что $B, B \to C \vdash C$. Поэтому по теореме о дедукции $B \vdash (B \to C) \to C$. В силу п. *a* имеем

$$(B \to C) \to C \vdash \overline{C} \to (\overline{B \to C})$$

(A заменено на $B \to C$, B — на C). Поэтому в силу свойства 4

$$B \vdash \overline{C} \to (\overline{B \to C}),$$

откуда соотношение *b* получается по теореме о дедукции.

Установим, наконец, справедливость утверждения c.

В силу п. a

$$A \to C \vdash \overline{C} \to \overline{A}, \quad \overline{A} \to C \vdash \overline{C} \to \overline{\overline{A}}.$$

Так как по условию $T \vdash A \to C$ и $T \vdash \overline{A} \to C$, то в силу свойства 4

$$T \vdash \overline{C} \to \overline{A} \ \text{ и } \ T \vdash \overline{C} \to \overline{\overline{A}},$$

а значит,

$$T, \overline{C} \vdash \overline{A} \ \text{ и } \ T, \overline{C} \vdash \overline{\overline{A}}.$$

В силу п. b

$$\overline{A}, \overline{\overline{A}} \vdash \overline{\overline{A} \to \overline{A}}.$$

Поэтому в силу свойства 4

$$T, \overline{C} \vdash \overline{\overline{A} \to \overline{A}},$$

откуда по теореме о дедукции

$$T \vdash \overline{C} \to (\overline{\overline{A} \to \overline{A}}).$$

Из схемы аксиом (3) получаем

$$\overline{C} \to (\overline{\overline{A} \to \overline{A}}) \vdash (\overline{A} \to \overline{A}) \to C.$$

Поэтому в силу свойства 4

$$T \vdash (\overline{A} \to \overline{A}) \to C.$$

В силу леммы 1 $\vdash \overline{A} \to \overline{A}$. Поэтому по правилу вывода $T \vdash C$.

Будем теперь считать, что формулам над $\{\overline{\ }, \to\}$ естественным образом сопоставлены функции алгебры логики. Всякую формулу, которая реализует тождественно истинную функцию, будем называть *тождественно истинной*.

Теорема 2. *Любая выводимая формула является тождественно истинной.*

Доказательство. Покажем сначала, что всякая формула, получающаяся по правилу вывода из тождественно истинных формул, является тождественно истинной. Это утверждение непосредственно следует из свойства импликации (табл.1).

Таблица 1

x	y	$x \to y$
0	0	1
0	1	1
1	0	0
1	1	1

В самом деле, если $x \to y = 1$ и $x = 1$, то $y = 1$, т. е. если A и $A \to B$ — тождественно истинные формулы, то B — тождественно истинная формула.

Покажем теперь, что каждая аксиома исчисления высказываний является тождественно истинной.

Схема аксиом (1). Рассмотрим формулу

$$A \to (B \to A),$$

где A и B — произвольные формулы. Если $A = 0$, то при любом значении B имеем $A \to (B \to A) = 1$; а если $A = 1$, то при любом значении B имеем $B \to A = 1$, и поэтому $A \to (B \to A) = 1$. Таким образом, $A \to (B \to A)$ — тождественно истинная формула.

Схема аксиом (2). Рассмотрим формулу

$$(A \to (B \to C)) \to ((A \to B) \to (A \to C)),$$

где A, B и C — произвольные формулы. Если $A = 0$, то $A \to C = 1$, а значит,

$$(A \to B) \to (A \to C) = 1,$$
$$(A \to (B \to C)) \to ((A \to B) \to (A \to C)) = 1$$

при любых значениях B и C. Если $A = 1$, то

$$A \to B = B, \ A \to C = C, \ A \to (B \to C) = B \to C.$$

Поэтому при любых значениях B и C выполняется равенство

$$(A \to (B \to C)) \to ((A \to B) \to (A \to C)) =$$
$$= (B \to C) \to (B \to C) = 1$$

Таким образом, формула $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ тождественно истинна.

Схема аксиом (3). Рассмотрим формулу

$$(\overline{A} \to \overline{B}) \to (B \to A),$$

где A и B — произвольные формулы. Если $A = 1$ или $B = 0$, то $B \to A = 1$ и $(\overline{A} \to \overline{B}) \to (B \to A) = 1$; если же $A = 0$ и $B = 1$, то $\overline{A} \to \overline{B} = 0$, и поэтому $(\overline{A} \to \overline{B}) \to (B \to A) = 1$. Таким образом, $(\overline{A} \to \overline{B}) \to (B \to A)$ — тождественно истинная формула.

Пусть A — выводимая формула. Докажем утверждение теоремы индукцией по длине n вывода формулы A. Если $n = 1$, т. е. вывод состоит из одной формулы, то A — аксиома, а значит, по доказанному выше является тождественно истинной формулой. Предположим, что все выводимые формулы, у которых длина вывода не превышает n, являются тождественно истинными. Пусть A — формула, вывод которой имеет длину $n + 1$:

$$A_1, A_2, \ldots, A_n, A_{n+1} =_\Gamma A.$$

Тогда либо A является аксиомой, а значит, по доказанному выше есть тождественно истинная формула, либо получается по правилу вывода из некоторых формул A_j и A_r (где $1 \leqslant j, r \leqslant n$), таких, что выполняется равенство $A_j =_\Gamma A_r \to A$. По индуктивному предположению формулы A_j и A_r являются тождественно истинными. Поэтому и формула A тождественно истинна.

Лекция № 15

Исчисление называется *противоречивым*, если найдется формула A, такая, что в этом исчислении одновременно выводимы формулы A и $\overline{A}$. В противном случае исчисление называется *непротиворечивым*.

Теорема 1. *Исчисление высказываний непротиворечиво.*

Доказательство. В самом деле, допустим, что формула A выводима. Тогда в силу теоремы 2 из предыдущей лекции она является тождественно истинной. Значит, $\overline{A}$ — тождественно ложная формула, и поэтому она не может быть выводимой.

Отметим, что из непротиворечивости исчисления высказываний следует существование невыводимых формул, например формула $\overline{A \to A}$ не является выводимой. С другой стороны, непротиворечивость исчисления высказываний следует из факта существования невыводимых формул. Действительно, пусть A — невыводимая формула. Предположим, что исчисление высказываний противоречиво. Тогда найдется формула B, такая, что $\vdash B$ и $\vdash \overline{B}$. В силу леммы 2, п. *а* из предыдущей лекции $\overline{B}, B \vdash A$. Поэтому $\vdash A$. Полученное противоречие показывает, что исчисление высказываний непротиворечиво.

Пусть $A(x_1, \ldots, x_n)$ — произвольная формула над множеством $\{\overline{x}, x \to y\}$, где $x_1, \ldots, x_n$ — все переменные, входящие в формулу A. Будем обозначать через A^ε формулу A, если $\varepsilon = 1$, и формулу $\overline{A}$, если $\varepsilon = 0$; $A(\alpha_1, \ldots, \alpha_n)$ — значение формулы A на наборе $(\alpha_1, \ldots, \alpha_n) \in E^n$ $(E = \{0, 1\})$.

Лемма. *Пусть $A(x_1, \ldots, x_n)$ — произвольная формула, $(\alpha_1, \ldots, \alpha_n)$ — произвольный набор переменных из E^n, а ε — значение $A(\alpha_1, \ldots, \alpha_n)$. Тогда*

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash A^\varepsilon.$$

Доказательство. Будем доказывать утверждение леммы индукцией по длине k формулы A. Если $k = 1$, то формула A имеет вид x_i, $1 \leqslant i \leqslant n$, и утверждение леммы $(x_i^{\alpha_i} \vdash x_i^{\alpha_i})$ очевидно.

Предположим, что утверждение леммы справедливо для формул, длина которых меньше k. Докажем его для формул длины k. Возможны два случая.

1. Формула A имеет вид $\overline{B}$. Тогда B имеет длину $k-1$. Положим $\varepsilon_1 = B(\alpha_1, \ldots, \alpha_n)$. Тогда $\varepsilon = \overline{\varepsilon_1}$. В силу индуктивного предположения

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash B^{\varepsilon_1}.$$

Если $\varepsilon_1 = 0$, то $\varepsilon = 1$ и

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash \overline{B},$$

т. е. $A^{\varepsilon} =_\Gamma \overline{B}$, и поэтому

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash A^{\varepsilon}.$$

Если $\varepsilon_1 = 1$, то $\varepsilon = 0$ и

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash B.$$

В силу леммы 2, п. c из предыдущей лекции $B \vdash \overline{\overline{B}}$. Поэтому

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash \overline{\overline{B}}.$$

Так как $A^{\varepsilon} =_\Gamma \overline{\overline{B}}$, то

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash A^{\varepsilon}.$$

Таким образом, в этом случае утверждение леммы доказано.

2. Формула A имеет вид $B \to C$. Тогда формулы B и C имеют длину меньше k. Положим $\varepsilon_1 = B(\alpha_1, \ldots, \alpha_n)$, $\varepsilon_2 = C(\alpha_1, \ldots, \alpha_n)$. Тогда $\varepsilon = \varepsilon_1 \to \varepsilon_2$. В силу индуктивного предположения

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash B^{\varepsilon_1},$$
$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash C^{\varepsilon_2}$$

(в тех случаях, когда формулы B или C содержат не все переменные $x_1, x_2, \ldots, x_n$, в левые части этих выражений добавляются лишние гипотезы).

Если $\varepsilon = 0$, то $\varepsilon_1 = 1$ и $\varepsilon_2 = 0$, т. е. $A^{\varepsilon} =_\Gamma (\overline{B \to C})$, и

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash B,$$
$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash \overline{C}.$$

Из леммы 3, п. b имеем $B, \overline{C} \vdash (\overline{B \to C})$. Таким образом, в силу свойства выводимости 4

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash (\overline{B \to C}).$$

Если $\varepsilon = 1$, $\varepsilon_1 = 0$, то $A^{\varepsilon} =_\Gamma B \to C$ и

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash \overline{B}.$$

Из леммы 2, п. a имеем $\overline{B} \vdash B \to C$, а значит,

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash A^{\varepsilon}.$$

Если $\varepsilon = 1$, $\varepsilon_1 = 1$, то $\varepsilon_2 = 1$, т. е. $A^{\varepsilon} =_\Gamma B \to C$ и

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash C.$$

В силу свойства выводимости 6

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash B \to C.$$

Таким образом, лемма полностью доказана.

Теорема 2 (о полноте)**.** *Всякая формула, выражающая тождественно истинную функцию, является выводимой.*

Доказательство. Пусть $A(x_1, \ldots, x_n)$ — произвольная тождественно истинная формула, а $(\alpha_1, \ldots, \alpha_n)$ — произвольный набор из E^n. В силу леммы

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_n^{\alpha_n} \vdash A.$$

Будем последовательно исключать гипотезы, пользуясь леммой 3, п. c из предыдущей лекции. При $\alpha_n = 1$ и $\alpha_n = 0$ имеем

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_{n-1}^{\alpha_{n-1}}, x_n \vdash A,$$
$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_{n-1}^{\alpha_{n-1}}, \overline{x}_n \vdash A.$$

Поэтому в силу леммы 3, п. c

$$x_1^{\alpha_1}, x_2^{\alpha_2}, \ldots, x_{n-1}^{\alpha_{n-1}} \vdash A.$$

Точно таким же образом, рассмотрев случаи, когда $\alpha_{n-1} = 1$ и $\alpha_{n-1} = 0$, применив лемму 3, п. c, исключим x_{n-1} и так далее; после шага $n-1$ получим

$$x_1 \vdash A,$$
$$\overline{x}_1 \vdash A,$$

откуда, применив лемму 3, п. c, окончательно получаем

$$\vdash A.$$

Таким образом, теорема доказана.

Покажем теперь, что добавление к множеству аксиом любой невыводимой формулы в качестве новой схемы аксиом приводит к противоречивости исчисления. В самом деле, пусть $D(x_1,\dots,x_n)$ — невыводимая формула. Рассмотрим новую систему аксиом, состоящую из схем аксиом (1)–(3), а также новой схемы

(4) $D(A_1,A_2,\dots,A_n)$,

где $A_1,A_2,\dots,A_n$ — произвольные формулы. Получим новое исчисление, в котором при выводе формул разрешается применять любую из схем аксиом (1)–(4). Выводимость формулы F в новом исчислении будем обозначать следующим образом: $\vdash_H F$.

Так как формула $D(x_1,\dots,x_n)$ невыводима в прежнем исчислении, то в силу теоремы о полноте она не равна тождественно 1. Поэтому найдется набор $(\alpha_1,\dots,\alpha_n)$ из E^n, такой, что

$$D(\alpha_1,\dots,\alpha_n)=0.$$

Обозначим через $F^{(1)}$ и $F^{(0)}$ формулы $x\to x$ и $\overline{x\to x}$ соответственно. Определим теперь формулы $F_1,F_2,\dots,F_n$ следующим образом. Положим $F_i =_\Gamma F^{(1)}$, если $\alpha_i=1$, и $F_i =_\Gamma F^{(0)}$, если $\alpha_i=0$. Легко видеть, что формула $D(F_1,\dots,F_n)$ — тождественно ложная формула. С другой стороны, формула $D(F_1,\dots,F_n)$ представляет собой частный случай схемы аксиом (4). Поэтому

$$\vdash_H D(F_1,\dots,F_n).$$

Рассмотрим теперь произвольную формулу B. Тогда формула $D(F_1,\dots,F_n)\to B$ является тождественно истинной. Поэтому в силу теоремы о полноте

$$\vdash D(F_1,\dots,F_n)\to B,$$

а значит, $\vdash_H B$.

Таким образом, в новом исчислении выводима любая формула, и, следовательно, оно противоречиво.

Независимость аксиом. Рассмотрим теперь следующий вопрос. Можно ли из схем аксиом (1)–(3) удалить какую-нибудь схему (изменив соответствующим образом понятие вывода) с сохранением полноты исчисления? То есть является ли какая-нибудь из этих схем аксиом зависимой от остальных?

Схема аксиом (i), $1\leqslant i\leqslant 3$, данного исчисления называется *независимой*, если найдется формула, которая представляет собой

частный случай схемы аксиом (i) и которую нельзя вывести с помощью правила вывода из остальных аксиом исчисления. Будем называть формулу (j,m)-*выводимой*, если при ее выводе используются только схемы аксиом (j) и (m), $1\leqslant j,m\leqslant 3$, $j\neq m$.

Для доказательства независимости схемы аксиом (i) используется следующий подход. Функции $\overline{x}$ и $x\to y$ рассматриваются как формулы k-значной логики ($k\geqslant 2$) и определяются некоторым специальным образом; кроме того, выбирается некоторое значение $\alpha\in E_k=\{0,1,\dots,k-1\}$. При этом задание этих функций и выбор α осуществляются таким образом, что выполняются следующие условия:

1) каждая формула, представляющая собой частный случай схем аксиом (j) или (m), $j,m\neq i$, тождественно равна α (т. е. реализует некоторую функцию k-значной логики, тождественно равную α);

2) правило вывода, примененное к формулам, тождественно равным α, дает формулу, тождественно равную α;

3) найдется формула, представляющая собой частный случай схемы аксиом (i), не равная тождественно α.

Утверждение 1. *Формула* $(\overline{x}\to\overline{y})\to(y\to x)$ *не является* $(1,2)$-*выводимой.*

Доказательство. Для доказательства этого утверждения будем рассматривать функции $\overline{x}$ и $x\to y$ как функции алгебры логики, определенные в соответствии с табл. 1 и 2.

Таблица 1

x	$\overline{x}$
0	0
1	1

$x\to y$:

Таблица 2

$x\setminus y$	0	1
0	1	1
1	0	1

Поскольку таблица значений функции $x\to y$ совпадает с прежним заданием импликации, то каждая формула, представляющая собой частный случай схем аксиом (1) или (2), тождественно равна 1. Правило вывода, примененное к формулам, тождественно равным 1, будет давать формулу, тождественно равную 1. Следовательно, каждая $(1,2)$-выводимая формула тождественно равна 1.

С другой стороны, формула $(\overline{x}\to\overline{y})\to(y\to x)$ не равна тождественно 1, поскольку при $x=0$, $y=1$ принимает значение

$(\overline{0} \to \overline{1}) \to (1 \to 0) = (0 \to 1) \to 0 = 1 \to 0 = 0$. Поэтому эта формула не является $(1,2)$-выводимой.

Утверждение 2. *Формула*

$$(x \to (y \to z)) \to ((x \to y) \to (x \to z))$$

не является $(1,3)$*-выводимой.*

Доказательство. Будем рассматривать функции $x \to y$ и $\overline{x}$ как функции 3-значной логики, определенные в соответствии с табл. 3 и 4.

Таблица 3

x	$\overline{x}$
0	1
1	0
2	2

$x \to y$:

Таблица 4

$x \setminus y$	0	1	2
0	1	1	1
1	0	1	2
2	2	1	1

Покажем, что каждая формула, представляющая собой частный случай схем аксиом (1) и (3), тождественно равна 1.

Схема аксиом (1). Рассмотрим формулу

$$A \to (B \to A),$$

где A и B — произвольные формулы. Если $A = 0$, то

$$A \to (B \to A) = 1$$

при любом значении B; если $A = 1$, то $B \to A = 1$ и

$$A \to (B \to A) = 1$$

при любом значении B; если $A = 2$, то

$$A \to (B \to A) = 2 \ \to \ (B \to 2) = 1$$

при любом значении B.

Таким образом, формула $A \to (B \to A)$ тождественно равна 1.

Схема аксиом (3). Рассмотрим формулу

$$(\overline{A} \to \overline{B}) \to (B \to A),$$

где A и B — произвольные формулы. Если $A = 1$, то $B \to A = 1$ и $(\overline{A} \to \overline{B}) \to (B \to A) = 1$. Если $A = 0$, то $(\overline{A} \to \overline{B}) = 1 \to \overline{B} = \overline{B}$,

а $(B \to A) = B \to 0 = \overline{B}$, и поэтому $(\overline{A} \to \overline{B}) \to (B \to A) = \overline{B} \to \overline{B} = 1$. Если $A = 2$, а $B \neq 1$, то $B \to A = 1$, а значит, $(\overline{A} \to \overline{B}) \to (B \to A) = 1$; если же $A = 2$, $B = 1$, то

$$(\overline{A} \to \overline{B}) \to (B \to A) = (\overline{2} \to \overline{1}) \to (1 \to 2) = \\ = (2 \to 0) \to 2 = 2 \to 2 = 1.$$

Таким образом, формула $(\overline{A} \to \overline{B}) \to (B \to A)$ тождественно равна 1.

Поскольку правило вывода, примененное к формулам, тождественно равным 1, будет давать формулу, тождественно равную 1, то каждая $(1,3)$-выводимая формула тождественно равна 1.

С другой стороны, формула

$$(x \to (y \to z)) \to ((x \to y) \to (x \to z))$$

не равна тождественно 1, поскольку при $x = y = 2$, $z = 0$ принимает значение

$$(2 \to (2 \to 0)) \to ((2 \to 2) \to (2 \to 0)) = \\ = (2 \to 2) \to (1 \to 2) = 1 \to 2 = 0.$$

Поэтому эта формула не является $(1,3)$-выводимой.

Утверждение 3. *Формула*

$$x \to (y \to x)$$

не является $(2,3)$*-выводимой.*

Доказательство. Будем рассматривать функции $x \to y$ и $\overline{x}$ как функции 3-значной логики, определенные в соответствии с табл. 5 и 6.

Таблица 5

x	$\overline{x}$
0	1
1	0
2	2

$x \to y$:

Таблица 6

$x \setminus y$	0	1	2
0	1	1	1
1	0	1	0
2	0	1	1

Покажем, что каждая формула, представляющая собой частный случай схем аксиом (2) и (3), тождественно равна 1.

Схема аксиом (2). Рассмотрим формулу

$$(A \to (B \to C)) \to ((A \to B) \to (A \to C)),$$

где A, B и C — произвольные формулы. Если $A = 0$, то $A \to C = 1$, а значит, $(A \to B) \to (A \to C) = 1$ при любых значениях B и C; если $C = 1$, то $A \to C = 1$ и $(A \to B) \to (A \to C) = 1$ при любых значениях A и B; если $A = C = 2$, то $A \to C = 1$ и

$$(A \to B) \to (A \to C) = 1$$

при любых значениях B; если $B = 0$, $A \in \{1, 2\}$ или если $B = 2$, $A = 1$, то $A \to B = 0$, а значит $(A \to B) \to (A \to C) = 1$. Итак, во всех перечисленных выше случаях $(A \to B) \to (A \to C) = 1$, а значит, формула

$$(A \to (B \to C)) \to (A \to B) \to (A \to C)$$

принимает значение 1.

Рассмотрим оставшиеся четыре случая:

1) $A = 1$, $B = 1$, $C = 0$;

2) $A = 1$, $B = 1$, $C = 2$;

3) $A = 2$, $B = 1$, $C = 0$;

4) $A = 2$, $B = 2$, $C = 0$.

Легко видеть, что в каждом из этих четырех случаев выполняются равенства

$$B \to C = 0, \quad A \to (B \to C) = 0.$$

Поэтому

$$(A \to (B \to C)) \to ((A \to B) \to (A \to C)) = 1.$$

Таким образом, рассматриваемая формула тождественно равна 1.

Схема аксиом (3). Рассмотрим формулу

$$(\overline{A} \to \overline{B}) \to (B \to A),$$

где A и B — произвольные формулы. Если $A = 1$, то $B \to A = 1$ и $(\overline{A} \to \overline{B}) \to (B \to A) = 1$ при любых значениях B; если $B = 0$, то $B \to A = 1$ и $(\overline{A} \to \overline{B}) \to (B \to A) = 1$ при любых значениях A; если $A = B = 2$, то $(\overline{A} \to \overline{B}) \to (B \to A) = 1$.

Рассмотрим оставшиеся три случая:

1) $A = 0$, $B = 1$;

2) $A = 0$, $B = 2$;

3) $A = 2$, $B = 1$.

Покажем, что в каждом из этих случаев значение формулы $\overline{A} \to \overline{B}$ равно 0. В самом деле, если $A = 0$, $B = 1$, то

$$\overline{A} \to \overline{B} = \overline{0} \to \overline{1} = 1 \to 0 = 0;$$

если $A = 0$, $B = 2$, то $\overline{A} \to \overline{B} = \overline{0} \to \overline{2} = 1 \to 2 = 0$; если $A = 2$, $B = 1$, то $\overline{A} \to \overline{B} = \overline{2} \to \overline{1} = 2 \to 0 = 0$. Так как в каждом из этих трех случаев $B \to A = 0$, то

$$(\overline{A} \to \overline{B}) \to (B \to A) = 0 \to 0 = 1.$$

Таким образом, формула $(\overline{A} \to \overline{B}) \to (B \to A)$ тождественно равна 1.

Поскольку правило вывода, примененное к формулам, тождественно равным 1, будет давать формулу, тождественно равную 1, то каждая $(2, 3)$-выводимая формула тождественно равна 1.

С другой стороны, формула $x \to (y \to x)$ не равна тождественно 1, поскольку при $x = 2$, $y = 1$ принимает значение $2 \to (1 \to 2) = 2 \to 0 = 0$. Поэтому эта формула не является $(2, 3)$-выводимой.

Из утверждений 1–3 следует

Теорема 3. *Каждая из схем аксиом* (1)–(3) *исчисления высказываний является независимой.*

ЛОГИКА ПРЕДИКАТОВ

Лекция № 16

Будем рассматривать предложения, зависящие от параметров. Например, "x — простое число", "$x \leqslant y$", "x делится на i". Каждое из этих предложений при конкретных значениях x, y, i становится высказыванием, принимающим значение "истина" или "ложь". Такого рода предложения называются предикатами. Более точно, *предикатами* будем называть функции $P(x_1, \ldots, x_n)$, аргументы которых принимают значения из некоторого множества M, а сами функции — значение 1 ("истина") или 0 ("ложь"). Предикат, зависящий от n переменных, называется n-местным предикатом. Предикат, переменные которого принимают значения из множества M, будем называть предикатом, определенным на множестве M (или предикатом над M). Например, предикат $P(x)$: "x — простое число" является одноместным предикатом, а предикат $S(x, y)$: "$x \leqslant y$" — двуместным. Для каждого натурального i можно рассмотреть одноместный предикат $P_i(x)$: "x делится на i"; в то же время предложение "x делится на i" является двуместным, если считать i переменной, принимающей натуральные значения.

Так же как и к высказываниям, к предикатам можно применять обычные логические операции (функции алгебры логики). В результате также будут получаться предикаты. Например, $P_2(x)\&P_3(x)$. Тем самым из некоторого исходного множества предикатов, используя операции $\&$, $\vee$, $\bar{}$, $\to$, мы можем составлять различные формулы, которые будут выражать некоторые предикаты.

Рассмотрим одноместные предикаты, определенные на некотором конечном множестве M. Система одноместных предикатов над M называется *полной*, если через них с помощью логических операций ($\&$, $\vee$, $\bar{}$, $\to$) можно выразить любой одноместный предикат над M.

Теорема. *Пусть M — конечное множество. Система $\{A_1(x), \ldots, A_s(x)\}$ одноместных предикатов над M является полной тогда и только тогда, когда для любых двух различных элементов a и b множества M найдется предикат A_i, $1 \leqslant i \leqslant s$, такой, что $A_i(a) \neq A_i(b)$.*

Доказательство. Предположим, что найдутся два различных элемента a и b из M, такие, что $A_i(a) = A_i(b)$ для всех $i = 1, \ldots, s$.

Тогда любая формула, составленная из предикатов $A_1(x), \ldots, A_s(x)$ с использованием операций $\&$, $\vee$, $\bar{}$, $\to$, будет выражать некоторый предикат над M с этим же свойством. Следовательно, нельзя получить, например, предикат, который принимает значение 1 на элементе a и значение 0 на всех остальных элементах множества M. Поэтому система $\{A_1(x), \ldots, A_s(x)\}$ не является полной.

Предположим теперь, что для любых двух различных элементов множества M найдется предикат из рассматриваемой системы, который принимает на них различные значения. Пусть $a \in M$. Положим

$$A_i^{(a)}(x) = \begin{cases} A_i(x), & \text{если } A_i(a) = 1; \\ \overline{A_i}(x), & \text{если } A_i(a) = 0, \end{cases}$$

$i = 1, \ldots, s$. Тогда $A_i(a) = 1$ для всех $i = 1, \ldots, s$. Рассмотрим предикат

$$A^{(a)}(x) = A_1^{(a)}(x)\&A_2^{(a)}(x)\& \ldots \&A_s^{(a)}(x).$$

Найдем значение этого предиката на элементах множества M. Очевидно, что $A^{(a)}(a) = 1$. Пусть $b \in M$, $b \neq a$. По условию существует предикат $A_j(x)$, $1 \leqslant j \leqslant s$, такой, что $A_j(b) \neq A_j(a)$. Легко видеть, что $A_j^{(a)}(b) \neq A_j^{(a)}(a)$, т. е. $A_j^{(a)}(b) = 0$. Поэтому $A^{(a)}(b) = 0$. Таким образом,

$$A^{(a)}(x) = \begin{cases} 1, & \text{если } x = a; \\ 0, & \text{если } x \neq a. \end{cases}$$

Пусть $P(x)$ — произвольный предикат над M. Если $P(x)$ — тождественно ложный предикат, то $P(x) = A_1(x)\&\overline{A_1(x)}$. В противном случае построим аналог совершенной дизъюнктивной нормальной формы. Пусть $a_1, \ldots, a_r$ — все элементы множества M, на которых предикат $P(x)$ принимает значение 1, $r \geqslant 1$. Тогда

$$P(x) = \bigvee_{i=1}^{r} A^{(a_i)}(x).$$

Таким образом, любой одноместный предикат над множеством M выражается через предикаты $A_1(x), \ldots, A_s(x)$ при помощи операций $\vee$, $\&$, $\bar{}$.

Определим две новые операции над предикатами — операции навешивания кванторов.

Квантор общности. Пусть $A(x, y_1, \ldots, y_n)$ — некоторый предикат, зависящий от переменных x, $y_1, \ldots, y_n$. Высказывание

"$A(x, y_1, \ldots, y_n)$ истинно для всех x" будем обозначать символом $(\forall x)A(x, y_1, \ldots, y_n)$ (читается "для всех x $A(x, y_1, \ldots, y_n)$"). Это высказывание зависит от переменных $y_1, \ldots, y_n$, причем на произвольном наборе $(\beta_1, \ldots, \beta_n)$ значений своих переменных оно принимает значение 1 тогда и только тогда, когда для любого значения α переменной x выполняется равенство $A(\alpha, \beta_1, \ldots, \beta_n) = 1$. Переход от предиката $A(x, y_1, \ldots, y_n)$ к предикату $(\forall x)A(x, y_1, \ldots, y_n)$ называется *навешиванием квантора общности*.

Квантор существования. Пусть $A(x, y_1, \ldots, y_n)$ — некоторый предикат. Высказывание "$A(x, y_1, \ldots, y_n)$ истинно при некотором x" будем обозначать символом $(\exists x)A(x, y_1, \ldots, y_n)$ (читается "существует x, для которого $A(x, y_1, \ldots, y_n)$"). Это высказывание зависит от переменных $y_1, \ldots, y_n$, причем на произвольном наборе $(\beta_1, \ldots, \beta_n)$ значений своих переменных оно принимает значение 1 тогда и только тогда, когда существует некоторое значение α переменной x, такое, что $A(\alpha, \beta_1, \ldots, \beta_n) = 1$. Переход от предиката $A(x, y_1, \ldots, y_n)$ к предикату $(\exists x)A(x, y_1, \ldots, y_n)$ называется *навешиванием квантора существования*.

Отметим, что применение каждой такой операции уменьшает число переменных, от которых зависит предикат, на единицу.

Примеры.

1. $(\forall x)P(x)$ — ложное высказывание.
2. $(\exists x)P_2(x)\&P_3(x)$ — истинное высказывание.
3. $(\forall x)(\forall y)(\forall z)(S(x, y)\&S(y, z) \to S(x, z))$ — истинное высказывание

(здесь $P(x)$, $P_2(x)$, $P_3(x)$ и $S(x, y)$ — предикаты из рассмотренных выше примеров).

Язык логики предикатов позволяет выражать более сложные предложения, чем язык логики высказываний.

Существует простая связь между логическими и теоретико-множественными операциями. Пусть $P(x_1, \ldots, x_n)$ — некоторый n-местный предикат над M. Обозначим через M_P множество всех наборов $(\alpha_1, \ldots, \alpha_n)$ из M^n, на которых предикат P принимает значение 1.

Пусть теперь $Q_1(x_1, \ldots, x_n)$ и $Q_2(x_1, \ldots, x_n)$ — некоторые n-местные предикаты над M, а M_{Q_1} и M_{Q_2} — множества, на которых предикаты Q_1 и Q_2 соответственно принимают значения 1. Тогда предикату $Q_1\&Q_2$ будет соответствовать пересечение множеств

M_{Q_1} и M_{Q_2}, предикату $Q_1 \vee Q_2$ будет соответствовать объединение этих множеств, а предикату $\overline{Q_1}$ — дополнение множества M_{Q_1}, т. е.

$$M_{Q_1\&Q_2} = M_{Q_1} \cap M_{Q_2},$$

$$M_{Q_1\vee Q_2} = M_{Q_1} \cup M_{Q_2},$$

$$M_{\overline{Q_1}} = M^n \setminus M_{Q_1}.$$

Предикату $(\exists x_1)Q_1(x_1, x_2, \ldots, x_n)$ соответствует проектирование множества M_{Q_1} вдоль оси x_1. В самом деле, если в M_{Q_1} входит хотя бы один набор $(\alpha_1, \alpha_2, \ldots, \alpha_n)$, то набор $(\alpha_2, \ldots, \alpha_n)$ принадлежит множеству $M_{(\exists x_1)Q_1}$.

Предикату $(\forall x_1)Q_1(x_1, x_2, \ldots, x_n)$ соответствует взятие дополнения к проекции вдоль оси x_1 дополнения множества M_{Q_1}, поскольку

$$\overline{(\forall x_1)Q_1(x_1, x_2, \ldots, x_n)} = (\exists x_1)\overline{Q_1(x_1, x_2, \ldots, x_n)}$$

(это равенство будет рассмотрено далее).

Определим более формально понятие формулы. Пусть M — некоторое множество. Множество M с определенными на нем предикатами

$$P_1^{(n_1)}(x_1, \ldots, x_{n_1}), P_2^{(n_2)}(x_1, \ldots, x_{n_2}), \ldots, P_k^{(n_k)}(x_1, \ldots, x_{n_k})$$

будем называть *моделью*. Множество $\Sigma = \{P_1^{(n_1)}, \ldots, P_k^{(n_k)}\}$ символов предикатов (с указанием числа переменных, от которых зависят предикаты), входящих в модель, будем называть *сигнатурой модели*.

Пусть $X = \{x_1, x_2, \ldots\}$ — счетное множество переменных, $\mathfrak{M} = \langle M, \Sigma\rangle$ — модель, Σ — сигнатура модели $\mathfrak{M}$. Определим по индукции следующие понятия:

a) формула в модели $\mathfrak{M}$;
b) множество X_F свободных переменных формулы F;
c) множество Y_F связанных переменных формулы F;
d) значение $F|_{\tilde{\alpha}}$ формулы F в модели $\mathfrak{M}$ на произвольном наборе $\tilde{\alpha}$ значений свободных переменных этой формулы.

1. Выражение $P_i^{(n_i)}(x_{j_1}, \ldots, x_{j_{n_i}})$ — формула; такие формулы будем называть *атомными*. Пусть $x_1, \ldots, x_s$ — все различные переменные этой формулы, $x_{j_1}, \ldots, x_{j_{n_i}} \in \{x_1, \ldots, x_s\}$. Множество

свободных переменных этой формулы есть множество $\{x_1, \ldots, x_s\}$; множество связанных переменных этой формулы пусто.

Пусть $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_s)$ — произвольный набор значений переменных из M^s. Тогда значение формулы $P_i^{(n_i)}(x_{j_1}, \ldots, x_{j_{n_i}})$ на наборе $\tilde{\alpha}$ равно $P_i^{(n_i)}(\alpha_{j_1}, \ldots, \alpha_{j_{n_i}})$.

2. Пусть F — формула, X_F — множество ее свободных переменных, Y_F — множество ее связанных переменных. Тогда $\overline{F}$ — формула; множество $X_{\overline{F}}$ ее свободных переменных есть X_F, множество $Y_{\overline{F}}$ ее связанных переменных есть Y_F. Значение этой формулы на любом наборе $\tilde{\alpha}$ свободных переменных противоположно соответствующему значению формулы F, т. е.

$$\overline{F}\big|_{\tilde{\alpha}} = \overline{F|_{\tilde{\alpha}}}.$$

3. Пусть F и G — формулы, X_F и X_G — множества свободных переменных формул F и G соответственно, а Y_F и Y_G — множества связанных переменных формул F и G соответственно, причем $X_F \cap Y_G = \emptyset$ и $X_G \cap Y_F = \emptyset$ (т. е. никакая свободная переменная формулы F не является связанной переменной формулы G и никакая свободная переменная формулы G не является связанной переменной формулы F). Тогда выражения

$$F \vee G, \quad F\&G, \quad F \to G$$

являются формулами. Множество свободных переменных каждой из них есть $X_F \cup X_G$; множество связанных переменных каждой из них есть $Y_F \cup Y_G$.

Пусть $\tilde{\alpha}$ — произвольный набор значений переменных из множества $X_F \cup X_G$. Этим набором определяется набор $\tilde{\beta}$ значений переменных из X_F и набор $\tilde{\gamma}$ значений переменных из X_G. Тогда

$$(F \vee G)|_{\tilde{\alpha}} = (F|_{\tilde{\beta}}) \vee (F|_{\tilde{\gamma}});$$
$$(F\&G)|_{\tilde{\alpha}} = (F|_{\tilde{\beta}}) \,\&\, (F|_{\tilde{\gamma}});$$
$$(F \to G)|_{\tilde{\alpha}} = (F|_{\tilde{\beta}}) \to (F|_{\tilde{\gamma}}).$$

4. Пусть F — формула, X_F и Y_F — множества ее свободных и связанных переменных соответственно, $x \in X_F$. Тогда $(\forall x)(F)$ — формула; множество ее свободных переменных есть $X_F \setminus \{x\}$, множество ее связанных переменных есть $Y_F \cup \{x\}$.

Пусть $X_F \setminus \{x\} = \{x_1, \ldots, x_t\}$, а $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_t)$ — произвольный набор значений переменных $(x_1, \ldots, x_t)$ (если x — единственная свободная переменная формулы F, то такой набор не берется). Значение $(\forall x)(F)|_{\tilde{\alpha}}$ формулы $(\forall x)(F)$ на наборе $\tilde{\alpha}$ определяется следующим образом. Рассматриваются значения формулы F на всех наборах $(\alpha_0, \alpha_1, \ldots, \alpha_t)$, где $\alpha_1, \ldots, \alpha_t$ фиксированы, а α_0 — произвольный элемент множества M. Если на всех таких наборах формула F принимает значение 1, то

$$(\forall x)\,(F)|_{\tilde{\alpha}} = 1.$$

Если же хотя бы для одного набора $(\alpha_0, \alpha_1, \ldots, \alpha_t)$ выполняется $F|_{(\alpha_0, \alpha_1, \ldots, \alpha_t)} = 0$, то

$$(\forall x)\,(F)|_{\tilde{\alpha}} = 0.$$

В формуле $(\forall x)\,(F)$ формула F называется *областью действия квантора* $(\forall x)$.

5. Пусть F — формула, X_F и Y_F — множества свободных и связанных переменных формулы F соответственно, $x \in X_F$. Тогда $(\exists x)(F)$ — формула; множество ее свободных переменных есть $X_F \setminus \{x\}$, множество ее связанных переменных есть $Y_F \cup \{x\}$.

Пусть $X_F \setminus \{x\} = \{x_1, \ldots, x_t\}$, а $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_t)$ — произвольный набор значений переменных $(x_1, \ldots, x_t)$ (если $X_F = \{x\}$, то такой набор не берется). Значение $(\exists x)\,(F)|_{\tilde{\alpha}}$ формулы $(\exists x)\,(F)$ на наборе $\tilde{\alpha}$ определяется следующим образом. Рассматриваются значения формулы F на всех наборах $(\alpha_0, \alpha_1, \ldots, \alpha_t)$, где $\alpha_1, \ldots, \alpha_t$ фиксированы, а α_0 — произвольный элемент множества M. Если на всех таких наборах формула F принимает значение 0, то

$$(\exists x)\,(F)|_{\tilde{\alpha}} = 0.$$

Если же хотя бы для одного набора $(\alpha_0, \alpha_1, \ldots, \alpha_t)$ выполняется $F|_{(\alpha_0, \alpha_1, \ldots, \alpha_t)} = 1$, то

$$(\exists x)\,(F)|_{\tilde{\alpha}} = 1.$$

В формуле $(\exists x)\,(F)$ формула F называется *областью действия квантора* $(\exists x)$.

Из определения формулы следует, что никакая переменная не может быть одновременно свободной и связанной.

Длиной формулы будем называть общее число входящих в нее знаков предикатов, кванторов и логических операций.

Введем некоторые соглашения, упрощающие записи формул (дополнительно к соглашениям, принятым в алгебре логики):

a) не будем заключать в скобки атомные формулы, а также формулы, в которых внешняя операция есть отрицание;

b) не будем заключать в скобки формулу, в которой внешняя операция есть навешивание квантора, если следующая операция также навешивание квантора;

c) будем считать, что квантор связывает сильнее всех других операций, и будем опускать соответствующие скобки.

Например, вместо

$$(\forall x)\Big((\forall y)\Big(\overline{(A(x,y)\to B(y,y))}\Big)\Big)$$

будем писать

$$(\forall x)(\forall y)\,\overline{A(x,y)\to B(y,y)}.$$

Сигнатурой формулы будем называть множество символов входящих в нее предикатов.

При фиксированной модели $\mathfrak{M}=\langle M,\Sigma\rangle$ каждая формула, имеющая свободные переменные, выражает некоторый предикат над M от этих переменных; формула, не имеющая свободных переменных, выражает некоторую константу (0 или 1).

Формула называется *истинной в модели* $\mathfrak{M}$, если она принимает значение 1 на всех наборах значений своих свободных переменных.

Пример. Пусть $P_2(x)$, $P_3(x)$, $P_5(x)$ и $P_6(x)$ — предикаты делимости на 2, 3, 5 и 6 соответственно, а $M=\{1,2,3,\dots\}$. Рассмотрим модель $\mathfrak{M}=\langle M,\Sigma\rangle$, где $\Sigma=\{P_2^{(1)},P_3^{(1)},P_5^{(1)},P_6^{(1)}\}$. Тогда формула

$$P_2(x)\&P_3(x)\to P_6(x)$$

истинна в модели $\mathfrak{M}$, а формула $P_2(x)\&P_5(x)\to P_6(x)$ не является истинной в $\mathfrak{M}$.

Пусть формулы F и G имеют одно и то же множество свободных переменных. Будем называть эти формулы *эквивалентными в модели* $\mathfrak{M}$, если на любом наборе значений свободных переменных они принимают равные значения (т. е. эти формулы выражают один и тот же предикат). Будем называть эти формулы *эквивалентными на множестве* M, если они эквивалентны во всех моделях $\mathfrak{M}=\langle M,\Sigma\rangle$, заданных на множестве M и имеющих сигнатуру Σ, включающую знаки предикатов этих формул. Будем называть формулы *эквивалентными*, если они эквивалентны на всех множествах.

Примеры.

1. Формулы $P_2(x)\&P_3(x)$ и $P_6(x)$ эквивалентны в модели $\mathfrak{M}$ из предыдущего примера.

   Рассмотрим теперь модель $\mathfrak{M}_1$, отличающуюся от модели $\mathfrak{M}$ тем, что предикат $P_6(x)$ задается иначе: $P_6(x)=0$ для всех $x\in M$. Тогда при $x=6$ первая формула принимает значение 1, а вторая — значение 0. Поэтому эти формулы не эквивалентны в модели $\mathfrak{M}_1$.

2. Рассмотрим формулы

   $$(\exists x)A(x)\ \text{ и }\ (\forall x)A(x).$$

   Легко видеть, что эти формулы эквивалентны на множестве из одного элемента. С другой стороны, существует определенная на множестве $\{a,b\}$ модель, в которой эти формулы не эквивалентны; для этого достаточно предикат $A(x)$ определить следующим образом:

   $$A(a)=0,\quad A(b)=1.$$

3. Очевидно, что формулы

   $$A(x)\ \text{ и }\ \overline{\overline{A}}(x)$$

   эквивалентны в любой модели, т. е. эквивалентны.

Лекция № 17

Рассмотрим правила преобразования формул, которые позволяют получать эквивалентные формулы, возможно, более удобного вида. Эквивалентность формул будем изображать при помощи знака равенства.

Легко видеть, что для формул логики предикатов справедливы все правила эквивалентных преобразований из алгебры логики. Справедлив и аналог правила замены на эквивалентную подформулу.

Имеются также правила, относящиеся к кванторам.

1. *Правило переименования связанных переменных.* В любой формуле при замене связанной переменной на другую переменную (в кванторе и всюду в области действия квантора) так, чтобы не нарушалось определение формулы, получается формула, эквивалентная исходной.

Это правило нетрудно доказать индукцией по длине формулы (с использованием правила замены на эквивалентную подформулу).

Пример. Формулы

$$(\forall x)(P(x,y)\vee Q(z)) \quad \text{и} \quad (\forall u)(P(u,y)\vee Q(z))$$

эквивалентны. В первой формуле связанной переменной является x, во второй — u; в обеих формулах свободными переменными являются y и z.

2. *Правило переноса квантора через отрицание.* Пусть $A(x)$ — формула, X_A — множество ее свободных переменных, $x\in X_A$. Тогда справедливы соотношения

$$\overline{(\forall x)A(x)} = (\exists x)\overline{A(x)},$$
$$\overline{(\exists x)A(x)} = (\forall x)\overline{A(x)}.$$

Докажем сначала первое из этих соотношений. Пусть $\mathfrak{M}$ — произвольная модель, сигнатура которой содержит знаки всех предикатов, входящих в формулу A.

Пусть $X_A=\{x,x_1,\ldots,x_t\}$, а $\tilde\alpha=(\alpha_1,\ldots,\alpha_t)$ — произвольный набор значений переменных $x_1,\ldots,x_t$ (если $X_A=\{x\}$, то такой набор не рассматривается). Рассмотрим все наборы $(\alpha_0,\alpha_1,\ldots,\alpha_t)$, где $\alpha_1,\ldots,\alpha_t$ фиксированы, а α_0 — произвольный элемент из множества M.

Если для любого элемента α_0

$$A(x)|_{(\alpha_0,\alpha_1,\ldots,\alpha_t)}=1,$$

то $(\forall x)A(x)|_{\tilde\alpha}=1$ и $\overline{(\forall x)A(x)|_{\tilde\alpha}}=0$. С другой стороны, так как $A(x)|_{(\alpha_0,\alpha_1,\ldots,\alpha_t)}=1$ для любого α_0, то $\overline{A(x)}|_{(\alpha_0,\alpha_1,\ldots,\alpha_t)}=0$ и поэтому

$$(\exists x)\overline{A}(x)|_{\tilde\alpha}=0.$$

Если же существует элемент α_0, для которого

$$A(x)|_{(\alpha_0,\alpha_1,\ldots,\alpha_t)}=0,$$

то $(\forall x)A(x)|_{\tilde\alpha}=0$ и $\overline{(\forall x)A(x)|_{\tilde\alpha}}=1$. С другой стороны, так как $A(x)|_{(\alpha_0,\alpha_1,\ldots,\alpha_t)}=0$, то $\overline{A(x)}|_{(\alpha_0,\alpha_1,\ldots,\alpha_t)}=1$ и поэтому

$$(\exists x)\overline{A}(x)|\tilde\alpha=1.$$

Таким образом, в обоих случаях значения этих формул равны.

Второе соотношение получается из первого при помощи правил эквивалентных преобразований функций алгебры логики. В самом деле, применим первое соотношение к формуле $\overline{A(x)}$. Тогда с учетом правила снятия двойного отрицания получим

$$\overline{(\forall x)\overline{A(x)}} = (\exists x)\overline{\overline{A(x)}} = (\exists x)A(x).$$

Поэтому

$$\overline{\overline{(\forall x)\overline{A(x)}}} = \overline{(\exists x)A(x)}$$

и, наконец,

$$(\forall x)\overline{A(x)} = \overline{(\exists x)A(x)}.$$

3. *Правило выноса квантора через скобки.* Пусть $A(x)$, B и $A(x)\vee B$ — формулы, X_A и X_B — множества свободных переменных формул A и B соответственно, $x\in X_A$ и формула B не содержит переменную x. Тогда

$$(\forall x)(A(x)\vee B)=(\forall x)A(x)\vee B.$$

В самом деле, пусть $x,x_1,x_2,\ldots,x_t$ — все свободные переменные формулы $A(x)\vee B$. Рассмотрим произвольный набор $\tilde\alpha=(\alpha_1,\ldots,\alpha_t)$ значений переменных $(x_1,\ldots,x_t)$. Этот набор определяет набор $\tilde\beta$ значений переменных из X_B. Так как формула B не содержит переменную x, то можно определить значение формулы B на наборе $\tilde\beta$.

Если $B|_{\tilde{\beta}} = 1$, то $((\forall x)A(x) \vee B)|_{\tilde{\alpha}} = 1$. С другой стороны, для любого набора $(\alpha_0, \alpha_1, \ldots, \alpha_t)$, где $\alpha_1, \ldots, \alpha_t$ фиксированы, а α_0 — произвольный элемент из множества M, имеем

$$(A(x) \vee B)|_{(\alpha_0,\alpha_1,\ldots,\alpha_t)} = 1.$$

Поэтому

$$(\forall x)(A(x) \vee B)|_{\tilde{\alpha}} = 1.$$

Если же $B|_{\tilde{\beta}} = 0$, то (так как B не содержит x)

$$(\forall x)(A(x) \vee B)|_{\tilde{\alpha}} = (\forall x)A(x)|_{\tilde{\alpha}} = ((\forall x)A(x) \vee B)|_{\tilde{\alpha}}.$$

При тех же условиях справедливы следующие равенства:

$$(\forall x)(A(x)\&B) = (\forall x)A(x)\&B;$$
$$(\exists x)(A(x) \vee B) = (\exists x)A(x) \vee B;$$
$$(\exists x)(A(x)\&B) = (\exists x)A(x)\&B.$$

Эти соотношения доказываются аналогичным образом.

Формулу будем называть *приведенной*, если в ней из логических операций встречаются лишь конъюнкция, дизъюнкция и отрицание, причем знак отрицания встречается только над атомными формулами.

Теорема 1. *Для любой формулы A существует эквивалентная ей приведенная формула B, причем множества связанных переменных формул A и B совпадают.*

Доказательство. Доказательство теоремы проведем индукцией по длине формулы.

Для формул длины 1 утверждение теоремы очевидно.

Предположим, что утверждение теоремы доказано для формул, длина которых меньше n. Докажем его для формул длины n. Пусть A — произвольная формула длины n. Она может быть одного из следующих видов:

1) $A_1 \vee A_2$;
2) $A_2\&A_2$;
3) $A_1 \to A_2$;
4) $(\forall x)A_1(x)$;
5) $(\exists x)A_1(x)$;
6) $\overline{A}_1$

(x — свободная переменная формулы $A_1(x)$).

Рассмотрим каждый из этих случаев отдельно.

1) Каждая из формул A_1, A_2 имеют длину менее n. Поэтому для них в силу индуктивного предположения существуют эквивалентные приведенные формулы B_1 и B_2 соответственно, причем $X_{A_i} = X_{B_i}$, $Y_{A_i} = Y_{B_i}$, $i = 1, 2$ (где X_{A_i} и Y_{A_i} — множества свободных и связанных переменных соответственно формулы A_i, а X_{B_i} и Y_{B_i} — множества свободных и связанных переменных соответственно формулы B_i, $i = 1, 2$). Так как $A_1 \vee A_2$ — формула, то $B_1 \vee B_2$ тоже формула. Она эквивалентна формуле A и является приведенной. Множество свободных переменных $(X_{B_1} \cup X_{B_2})$ формулы $B_1 \vee B_2$ совпадает с множеством $(X_{A_1} \cup X_{A_2})$ свободных переменных формулы A; множество $(Y_{B_1} \cup Y_{B_2})$ связанных переменных формулы $B_1 \vee B_2$ совпадает с множеством $(Y_{A_1} \cup Y_{A_2})$ связанных переменных формулы A.

2) Этот случай рассматривается аналогично предыдущему.

3) В этом случае формула A эквивалентна формуле $\overline{A}_1 \vee A_2$. Так как каждая из формул A_1, A_2 имеет длину менее $n-1$, то формула $\overline{A}_1$ имеет длину менее n. Поэтому по индуктивному предположению для формул $\overline{A}_1$ и A_2 существуют эквивалентные приведенные формулы B_1 и B_2 соответственно. Тогда в качестве приведенной формулы, эквивалентной формуле A, возьмем формулу $B_1 \vee B_2$.

4) Формула $A_1(x)$ имеет длину менее n. Поэтому для нее по индуктивному предположению существует эквивалентная приведенная формула $B(x)$. В качестве приведенной формулы, эквивалентной формуле A, возьмем формулу $(\forall x)B(x)$.

5) Этот случай рассматривается аналогично предыдущему.

6) Этот случай распадается на несколько подслучаев в зависимости от вида формулы A_1.

Если выполняется равенство $A_1 =_\Gamma C_1 \vee C_2$, то A эквивалентна формуле $\overline{C}_1\&\overline{C}_2$; если $A_1 =_\Gamma C_1\&C_2$, то A эквивалентна формуле $\overline{C}_1 \vee \overline{C}_2$; если $A_1 =_\Gamma C_1 \to C_2$, то A эквивалентна формуле $C_1\&\overline{C}_2$; если $A_1 =_\Gamma (\forall x)C(x)$, то A эквивалентна формуле $(\exists x)\overline{C(x)}$; если $A_1 =_\Gamma (\exists x)C(x)$, то A эквивалентна формуле $(\forall x)\overline{C(x)}$; наконец, если $A_1 =_\Gamma \overline{C}$, то A эквивалентна формуле C.

Формулы $C_1, C_2, \overline{C}_1, \overline{C}_2, C(x), C$ имеют длину менее n. Поэтому для каждой из них по индуктивному предположению существует эквивалентная приведенная формула. Дальнейшие рассуждения для каждого из этих подслучаев аналогичны рассуждениям, приведенным выше для случаев 1–5 (в последнем подслучае достаточно взять приведенную формулу, эквивалентную формуле C).

Теорема полностью доказана.

Приведенная формула называется *нормальной*, если или она не содержит кванторов, или в ней операции взятия кванторов следуют за всеми другими операциями. Другими словами, нормальная формула со свободными переменными $x_1, \ldots, x_k$ и связанными переменными $y_1, \ldots, y_s$ имеет следующий вид:

$$(Q_1 y_1)(Q_2 y_2) \ldots (Q_s y_s) A(x_1, \ldots, x_k, y_1, \ldots, y_s),$$

где $Q_i \in \{\forall, \exists\}$, $i = 1, \ldots, s$, а $A(x_1, \ldots, x_k, y_1, \ldots, y_s)$ — приведенная формула, не содержащая кванторов.

Теорема 2. *Для любой формулы A существует эквивалентная ей нормальная формула.*

Доказательство. Пусть A — произвольная формула. В силу теоремы 1 существует эквивалентная ей приведенная формула B. Пусть X_B и Y_B — множество свободных и множество связанных переменных формулы B соответственно. Если B не содержит кванторов, то она является нормальной и утверждение теоремы доказано.

Пусть формула B содержит k кванторов $(Q_1 y_{i_1}), \ldots, (Q_k y_{i_k})$, где $Q_1, \ldots, Q_k \in \{\forall, \exists\}$, $y_{i_1}, \ldots, y_{i_k} \in Y_B$; $k \geqslant |Y_B|$.

Выберем k новых различных переменных $z_1, \ldots, z_k \notin X_B \cup Y_B$ и при помощи правила переименования связанных переменных (в кванторе и всюду в области действия квантора) заменим переменные $y_{i_1}, \ldots, y_{i_k}$ на переменные $z_1, \ldots, z_k$ соответственно. В результате получим формулу C. Она эквивалентна формуле B. Множество X_C свободных переменных этой формулы совпадает с множеством X_B; множество Y_C связанных переменных есть множество $\{z_1, \ldots, z_k\}$.

Далее, применяя правила выноса квантора через скобки, выносим все кванторы наружу, после чего формула принимает нормальный вид. Более точно, если в формуле C имеется некоторая подформула C_1 вида $(Qz)D(z) \circ E$ (или $E \circ (Qz)D(z)$), где $Q \in \{\forall, \exists\}$, $\circ \in \{\vee, \&\}$, а $z \in \{z_1, \ldots, z_k\}$, то заменим C_1 на эквивалентную ей в силу правила 3 формулу $(Qz)(D(z) \circ E)$ (поскольку формула E не содержит z). Нетрудно убедиться в том, что, последовательно применяя необходимое число раз данное эквивалентное преобразование, можно привести формулу C к нормальному виду.

Пример.

$$(\forall x)A(x) \vee (\exists x)B(x) = (\forall z_1)A(z_1) \vee (\exists z_2)B(z_2) =$$
$$= (\forall z_1)(A(z_1) \vee (\exists z_2)B(z_2)) = (\forall z_1)(\exists z_2)(A(z_1) \vee B(z_2)).$$

Пусть формула A имеет сигнатуру Σ. Будем называть эту формулу *истинной на множестве M*, если она истинна во всех моделях, определенных на множестве M и имеющих сигнатуру Σ.

Формула называется *тождественно истинной* (или *общезначимой*), если она истинна на всех множествах (или, что то же самое, если она истинна во всех моделях).

Примеры.

1. Формула $(\exists x)A(x) \to (\forall x)A(x)$ является истинной на одноэлементных множествах и не является истинной на множествах из большего числа элементов.
2. Формула
$$((\forall x_1)(\exists x_2)\overline{A(x_1, x_2)} \& (\forall y_2)(\exists y_1)A(y_1, y_2)) \to$$
$$\to ((\forall z_1)A(z_1, z_1) \vee (\forall z_2)\overline{A(z_2, z_2)})$$
является истинной на множествах, имеющих не более двух элементов, и не является истинной на множествах из трех и более элементов.
3. Формула
$$(\exists x)(\forall y)A(x, y) \to (\forall y)(\exists x)A(x, y)$$
является тождественно истинной.

Задача установления тождественной истинности формул логики предикатов является существенно более сложной, чем для формул алгебры логики. В общем случае вообще не существует алгоритма для решения этой задачи. Однако для некоторых частных случаев эффективное решение возможно, например для случая, когда все предикаты в формулах являются одноместными.

Лекция № 18

Рассмотрим задачу установления тождественной истинности формул, сигнатура которых состоит только из одноместных предикатных символов.

Пусть $\Sigma = \{P_1^{(1)}, \ldots, P_K^{(1)}\}$ — множество одноместных предикатных символов, а $\mathfrak{M}_1 = \langle M_1, \Sigma\rangle$ и $\mathfrak{M}_2 = \langle M_2, \Sigma\rangle$ — две модели сигнатуры Σ, определенные на множествах M_1 и M_2 соответственно. Будем говорить, что модель $\mathfrak{M}_1$ *гомоморфна* модели $\mathfrak{M}_2$, если существует функция $\varphi(x)$, отображающая множество M_1 на множество[1)] M_2, такое, что для любого элемента a из M_1 и любого предиката $P_i^{(1)}(x)$ $(1 \leqslant i \leqslant k)$ выполняется равенство

$$P_i^{(1)}(a) = P_i^{(1)}(\varphi(a))$$

(в левой части равенства стоит предикат в модели $\mathfrak{M}_1$, а в правой — в модели $\mathfrak{M}_2$).

Лемма. *Пусть модель $\mathfrak{M}_1 = \langle M_1, \Sigma\rangle$ гомоморфна модели $\mathfrak{M}_2 = \langle M_2, \Sigma\rangle$, F — произвольная формула сигнатуры Σ, $\{x_1, \ldots, x_n\}$ — множество свободных переменных формулы F, $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ — произвольный набор значений переменных $(x_1, \ldots, x_n)$, а $\varphi(\tilde{\alpha}) = (\varphi(\alpha_1), \ldots, \varphi(\alpha_n))$. Тогда*

$$F|_{\tilde{\alpha}} = F|_{\varphi(\tilde{\alpha})},$$

где $F|_{\tilde{\alpha}}$ — значение формулы F на наборе $\tilde{\alpha}$ в модели $\mathfrak{M}_1$, а $F|_{\varphi(\tilde{\alpha})}$ — значение формулы F на наборе $\varphi(\tilde{\alpha})$ в модели $\mathfrak{M}_2$.

Доказательство. В силу теоремы 2 из предыдущей лекции можно считать, что F — нормальная формула.

Рассмотрим сначала случай, когда F не содержит кванторов. Докажем утверждение леммы индукцией по длине l формулы F.

Для формул длины 1 и 2 (т. е. формул вида $P_{i_1}^{(1)}(x_{j_1})$ и $\overline{P_{i_2}^{(1)}}(x_{j_2})$) утверждение леммы следует из определения гомоморфизма моделей $\mathfrak{M}_1$ и $\mathfrak{M}_2$.

Предположим, что утверждение леммы справедливо для нормальных формул, которые не содержат кванторов и имеют длину менее l. Пусть F — нормальная формула без кванторов длины l, $l \geqslant 2$. Тогда она имеет один из следующих видов:

[1)] То есть каждому элементу a из M_1 соответствует единственный элемент $\varphi(a)$ из M_2, и для каждого элемента b из M_2 существует некоторый (необязательно единственный) элемент a из M_1, такой, что $\varphi(a) = b$.

1) $F_1 \vee F_2$;
2) $F_1 \& F_2$.

Рассмотрим первый случай. Пусть $X_F = \{x_1, x_2, \ldots, x_n\}$ — множество свободных переменных формулы F, а $\{x_{i_1}, \ldots, x_{i_r}\}$ и $\{x_{j_1}, \ldots, x_{j_q}\}$ — множества свободных переменных формул F_1 и F_2 соответственно. Рассмотрим произвольный набор $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ значений переменных $(x_1, \ldots, x_n)$ в модели $\mathfrak{M}_1$. Набор $\tilde{\alpha}$ однозначно определяет наборы $\tilde{\beta} = (\alpha_{i_1}, \ldots, \alpha_{i_r})$ и $\tilde{\gamma} = (\alpha_{j_1}, \ldots, \alpha_{j_q})$ значений переменных $(x_{i_1}, \ldots, x_{i_r})$ и $(x_{j_1}, \ldots, x_{j_q})$ соответственно в модели $\mathfrak{M}_1$. Положим $\varphi(\tilde{\alpha}) = (\varphi(\alpha_1), \ldots, \varphi(\alpha_n))$, $\varphi(\tilde{\beta}) = (\varphi(\alpha_{i_1}), \ldots, \varphi(\alpha_{i_r}))$, $\varphi(\tilde{\gamma}) = (\varphi(\alpha_{j_1}), \ldots, \varphi(\alpha_{j_q}))$.

По индуктивному предположению

$$F_1|_{\tilde{\beta}} = F_1|_{\varphi(\tilde{\beta})}, \quad F_2|_{\tilde{\gamma}} = F_2|_{\varphi(\tilde{\gamma})}.$$

Поэтому

$$F|_{\tilde{\alpha}} = (F_1 \vee F_2)|_{\tilde{\alpha}} = F_1|_{\tilde{\beta}} \vee F_2|_{\tilde{\gamma}} =$$
$$= F_1|_{\varphi(\tilde{\beta})} \vee F_2|_{\varphi(\tilde{\gamma})} = (F_1 \vee F_2)|_{\varphi(\tilde{\alpha})} = F|_{\varphi(\tilde{\alpha})}.$$

Второй случай рассматривается аналогично.

Докажем теперь утверждение леммы для произвольных нормальных формул индукцией по числу m кванторов. Для случая $m = 0$ это утверждение было доказано выше.

Предположим, что утверждение леммы справедливо для формул, содержащих менее m кванторов. Докажем его для формул с m кванторами. Пусть F — нормальная формула с m кванторами. Тогда она имеет один из следующих видов:

a) $(\forall x)F_1(x)$;
b) $(\exists x)F_1(x)$

(где F_1 — формула с $m-1$ кванторами, а x — свободная переменная формулы F_1).

Рассмотрим случай *a*. Пусть $X_{F_1} = \{x, x_1, \ldots, x_n\}$ — множество свободных переменных формулы F_1. Рассмотрим произвольный набор $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ значений переменных $(x_1, x_2, \ldots, x_n)$ в модели $\mathfrak{M}_1$ (если $X_{F_1} = \{x\}$, то такой набор не рассматривается). Рассмотрим также все наборы $(\alpha_0, \alpha_1, \ldots, \alpha_n)$, где $\alpha_1, \ldots, \alpha_n$ фиксированы, а α_0 — произвольный элемент из M_1. Положим

$\varphi(\tilde{\alpha}) = (\varphi(\alpha_1), \ldots, \varphi(\alpha_n))$. В силу индуктивного предположения для любого элемента α_0 выполняется равенство

$$F_1|_{(\alpha_0,\alpha_1,\ldots,\alpha_n)} = F_1|_{(\varphi(\alpha_0),\varphi(\alpha_1),\ldots,\varphi(\alpha_n))}. \qquad (*)$$

Если для любого α_0 из M_1

$$F_1|_{(\alpha_0,\alpha_1,\ldots,\alpha_n)} = 1,$$

то в силу соотношения $(*)$ и для любого элемента β_0 из M_2

$$F_1|_{(\beta_0,\varphi(\alpha_1),\ldots,\varphi(\alpha_n))} = 1.$$

Поэтому

$$F|_{\tilde{\alpha}} = (\forall x)F_1|_{\tilde{\alpha}} = 1,$$
$$F|_{\varphi(\tilde{\alpha})} = (\forall x)F_1|_{\varphi(\tilde{\alpha})} = 1.$$

Если для некоторого элемента α_0 из M_1

$$F_1|_{(\alpha_0,\alpha_1,\ldots,\alpha_n)} = 0,$$

то в силу $(*)$

$$F_1|_{(\varphi(\alpha_0),\varphi(\alpha_1),\ldots,\varphi(\alpha_n))} = 0.$$

Поэтому

$$F|_{\tilde{\alpha}} = (\forall x)F_1|_{\tilde{\alpha}} = 0,$$
$$F|_{\varphi(\tilde{\alpha})} = (\forall x)F_1|_{\varphi(\tilde{\alpha})} = 0.$$

Случай b рассматривается аналогично предыдущему.

Теорема 1. *Пусть модель $\mathfrak{M}_1$ содержит только одноместные предикаты $P_1^{(1)}(x), \ldots, P_n^{(1)}(x)$. Тогда существует модель $\mathfrak{M}_2$ той же сигнатуры, определенная на некотором множестве из q элементов, $q \leqslant 2^n$, обладающая следующим свойством. Любая формула F является истинной в модели $\mathfrak{M}_1$ тогда и только тогда, когда она истинна в модели $\mathfrak{M}_2$.*

Доказательство. Пусть $\mathfrak{M}_1 = \langle M_1, \Sigma\rangle$, где Σ — сигнатура модели $\mathfrak{M}_1$, $\Sigma = \{P_1^{(1)}, \ldots, P_n^{(1)}\}$, а M_1 — множество, на котором определена модель $\mathfrak{M}_1$. Разобьем множество M_1 на 2^n классов $K_{\alpha_1,\alpha_2,\ldots,\alpha_n}$, где $\alpha_1, \alpha_2, \ldots, \alpha_n \in \{0, 1\}$, следующим образом:

$$a \in K_{P_1^{(1)}(a),P_2^{(1)}(a),\ldots,P_n^{(1)}(a)},$$

т. е. каждый элемент принадлежит классу, у которого набор индексов есть набор значений предикатов $P_1^{(1)}(x), \ldots, P_n^{(1)}(x)$ на этом элементе. При этом некоторые из классов могут оказаться пустыми. Пусть q — число непустых классов, $q \geqslant 1$. Определим множество M_2 следующим образом. Выберем в каждом непустом классе по одному элементу; пусть $a_1, \ldots, a_q$ — выбранные элементы. Положим

$$M_2 = \{a_1, \ldots, a_q\}.$$

Рассмотрим модель $\mathfrak{M}_2 = \langle M_1, \Sigma\rangle$, в которой значение $P_i^{(1)}(a)$ каждого предиката $P_i^{(1)}(x)$ на любом элементе a из M_2 совпадает со значением предиката $P_i^{(1)}(x)$ на элементе a в модели $\mathfrak{M}_1$, $1 \leqslant i \leqslant n$.

Покажем, что модель $\mathfrak{M}_2$ гомоморфна модели $\mathfrak{M}_1$. Определим функцию φ следующим образом. Пусть a — некоторый элемент из M_1. Он принадлежит некоторому классу $K_{\alpha_1,\ldots,\alpha_n}$. Пусть a_i — элемент, выбранный из этого класса ($a_i \in M_2$). Положим

$$\varphi(a) = a_i.$$

Так как $a \in K_{\alpha_1,\ldots,\alpha_n}$ и $a_i \in K_{\alpha_1,\ldots,\alpha_n}$, то для всех $j = 1, \ldots, n$ выполняется равенство $P_j^{(1)}(a) = P_j^{(1)}(a_i)(= \alpha_j)$. Поэтому для каждого элемента a из M_1 и любого предиката $P_j^{(1)}(x)$, где $1 \leqslant j \leqslant n$, выполняется равенство

$$P_j^{(1)}(a) = P_j^{(1)}(\varphi(a)).$$

Таким образом, модель $\mathfrak{M}_1$ гомоморфна модели $\mathfrak{M}_2$.

Пусть F — произвольная формула в модели $\mathfrak{M}_1$. В силу леммы F истинна в модели $\mathfrak{M}_1$ тогда и только тогда, когда она истинна в модели $\mathfrak{M}_2$ (гомоморфной модели $\mathfrak{M}_1$).

Из теоремы 1 следует

Теорема 2. *Пусть F — произвольная формула сигнатуры $\Sigma = \{P_1^{(1)}, \ldots, P_n^{(1)}\}$. Для того чтобы F была тождественно истинной, необходимо и достаточно, чтобы она была тождественно истинной на всех моделях, определенных на множествах, состоящих не более чем из 2^n элементов.*

Нетрудно убедиться в том, что достаточно рассмотреть модели, которые определены на множествах, содержащих ровно 2^n элементов.

Отметим, что теорема 2 дает эффективный способ проверки тождественной истинности формул, содержащих только одноместные предикаты.

Пусть $M = \{b_1, \ldots, b_k\}$ — некоторое конечное множество, $\mathfrak{M} = \langle M, \Sigma\rangle$ — модель сигнатуры Σ, $F(x)$ — формула в модели $\mathfrak{M}$, $X_F = \{x, x_1, \ldots, x_n\}$ — множество свободных переменных формулы F, а $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ — произвольный набор значений переменных $(x_1, \ldots, x_n)$ в модели $\mathfrak{M}$. Легко видеть, что выполняются следующие соотношения:

$$(\exists x)F(x)|_{\tilde{\alpha}} = F(x)|_{(b_1,\alpha_1,\ldots,\alpha_n)} \vee \cdots \vee F(x)|_{(b_k,\alpha_1,\ldots,\alpha_n)};$$

$$(\forall x)F(x)|_{\tilde{\alpha}} = F(x)|_{(b_1,\alpha_1,\ldots,\alpha_n)} \& \ldots \& F(x)|_{(b_k,\alpha_1,\ldots,\alpha_n)}.$$

Кроме того, формула $F(x, x_1, \ldots, x_n)$, где $\{x, x_1, \ldots, x_n\}$ — множество свободных переменных формулы F, является истинной на множестве M тогда и только тогда, когда истинна на множестве M формула

$$(\forall x)(\forall x_1)\ldots(\forall x_n)\, F(x, x_1, \ldots, x_n).$$

Таким образом, проверка тождественной истинности формул, сигнатура которых содержит только одноместные предикатные символы, сводится к проверке тождественной истинности формул алгебры логики. При этом достаточно рассматривать формулы, не содержащие свободных переменных.

Пример. Рассмотрим формулу F следующего вида:

$$(\forall x)(\exists y)\,(P_1^{(1)}(x) \vee P_2^{(1)}(y)),$$

где $P_1^{(1)}$ и $P_2^{(1)}$ — одноместные предикатные символы. По теореме 2 для проверки тождественной истинности формулы F достаточно исследовать модели $\mathfrak{M} = \langle M, \Sigma\rangle$, такие, что $|M| = 2^2 = 4$, $\Sigma = \{P_1^{(1)}, P_2^{(1)}\}$. Пусть $M = \{a_1, a_2, a_3, a_4\}$. Тогда

$$(\forall x)(\exists y)\,(P_1^{(1)}(x) \vee P_2^{(1)}(y)) =$$

$$= (\forall x)\,((P_1^{(1)}(x) \vee (\exists y)P_2^{(1)}(y)) =$$

$$= P_1^{(1)}(a_1)\,\&\,P_1^{(1)}(a_2)\,\&\,P_1^{(1)}(a_3)\,\&\,P_1^{(1)}(a_4) \vee$$

$$\vee(P_2^{(1)}(a_1) \vee P_2^{(1)}(a_2) \vee P_2^{(1)}(a_3) \vee P_2^{(1)}(a_4)).$$

Обозначим $P_i^{(1)}(a_j)$ через x_{ij} $(1 \leqslant i \leqslant 2,\ 1 \leqslant j \leqslant 4)$. Получим формулу

$$x_{11}x_{12}x_{13}x_{14} \vee x_{21} \vee x_{22} \vee x_{23} \vee x_{24},$$

которая выражает некоторую функцию алгебры логики

$$f(x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24})$$

от восьми переменных. По теореме 2 формула F тождественно истинна тогда и только тогда, когда она тождественно истинна во всех моделях, определенных на множестве из четырех элементов. Поэтому F является тождественно истинной тогда и только тогда, когда функция f тождественно равна единице. Поскольку $f(0, 0, \ldots, 0) = 0$, то формула F не является тождественно истинной.

ИСЧИСЛЕНИЕ ПРЕДИКАТОВ

Лекция № 19

Все тождественно истинные формулы логики предикатов (так же, как это было в логике высказываний) могут быть получены при помощи некоторого исчисления. Как и в исчислении высказываний, указываются некоторое множество исходных формул и некоторые правила образования по одним формулам других. Так же, как и ранее, мы ограничимся рассмотрением формул, в которых используются только две логические операции: импликация и отрицание.

Исходными формулами будут формулы следующих пяти типов:

(1) $A \to (B \to A)$;

(2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$;

(3) $(\overline{A} \to \overline{B}) \to (B \to A)$;

(4) $(\forall x)A(x) \to A(y)$;

(5) $A(x) \to (\exists y)A(y)$,

при этом в первые три выражения вместо A, B и C могут подставляться любые формулы логики предикатов, такие, что выражения (1)–(3) также являются формулами (т. е. выполнены все требования для свободных и связанных переменных); выражения (4) и (5) являются формулами, и в этих выражениях $A(x)$ — формула, у которой x — свободная переменная, а формула $A(y)$ получается из $A(x)$ заменой всех вхождений переменной x на переменную y.

Формулы этих типов будем называть *аксиомами*. Сами эти выражения называются также *схемами аксиом*, так как каждое из них (как и в исчислении высказываний) определяет бесконечное множество различных формул.

Рассмотрим следующие правила вывода:

I. $\dfrac{A \to B, A}{B}$.

II. $\dfrac{B \to A(x)}{B \to (\forall x)A(x)}$, где формула B не содержит x.

III. $\dfrac{A(x) \to B}{(\exists x)A(x) \to B}$, где формула B не содержит x.

IV. Переименование свободной переменной: свободную переменную, входящую в формулу A, можно заменить (во всех вхождениях) другой переменной, не являющейся связанной в A.

V. Переименование связанной переменной: связанную переменную формулы A можно заменить другой переменной (во всех вхождениях в области действия квантора и в кванторе) так, чтобы получившееся выражение было формулой.

В правилах I–III будем говорить о формуле, стоящей под чертой, что она получена из формулы (формул), стоящей над чертой, по соответствующему правилу. В правилах IV и V будем говорить о формуле, которая получена в результате описанной в правиле процедуры, что она получена из формулы A по соответствующему правилу.

Выводом называется конечная последовательность формул

$$F_1, F_2, \ldots, F_k,$$

где F_i $(1 \leqslant i \leqslant k)$ либо является аксиомой, либо получена из предыдущих формул этой последовательности по одному из правил вывода I–V. Формула F называется *выводимой*, если существует вывод этой формулы, т. е. существует вывод $F_1, F_2, \ldots, F_k$, в котором формула F_k совпадает с F. Выводимость формулы F (как и в исчислении высказываний) будем обозначать следующим образом: $\vdash F$.

В исчислении предикатов можно определить понятие вывода из системы гипотез. Однако это понятие является более сложным, чем в исчислении высказываний, и здесь использоваться не будет.

Приведем ослабленный вариант этого понятия.

Пусть $T = \{A_1, \ldots, A_n\}$ — конечное (быть может, пустое) множество формул. *Специальным выводом* из системы гипотез T называется конечная последовательность формул

$$F_1, F_2, \ldots, F_k,$$

где F_i $(1 \leqslant i \leqslant k)$ является либо аксиомой (одной из (1)–(5)), либо одной из формул системы T, либо получена по правилу I из некоторых формул F_j и F_r этой последовательности с меньшими номерами $j, r < i$.

Формула F называется *специально выводимой* из системы гипотез T, если существует специальный вывод $F_1, F_2, \ldots, F_k$ из системы гипотез T, в котором формула F_k совпадает с F. Специальную выводимость формулы F будем обозначать следующим образом: $\vdash_C F$.

Очевидно, что специальный вывод из пустой системы гипотез есть просто вывод. Поэтому если $\vdash_C F$, то $\vdash F$.

В исчислении предикатов справедлива также (в более сложной формулировке) теорема о дедукции. Приведем ослабленный вариант этой теоремы.

Теорема 1. *Пусть T — конечное множество формул, A и B — формулы. Если $T, A \vdash_C B$, то $T \vdash_C A \to B$.*

Доказательство этой теоремы дословно совпадает с доказательством теоремы о дедукции для исчисления высказываний.

В силу этой теоремы все полученные ранее утверждения о выводимости формул в исчислении высказываний можно рассматривать как утверждения о специальной выводимости формул исчисления предикатов.

Лемма 1. *Каждая аксиома исчисления предикатов является тождественно истинной формулой.*

Доказательство. Для схем аксиом (1)–(3) утверждение леммы следует из тождественной истинности аксиом для исчисления высказываний.

Схема аксиом (4). Рассмотрим формулу

$$(\forall x)\, A(x) \to A(y).$$

Пусть $\{x, x_1, \ldots, x_n\}$ — множество свободных переменных формулы $A(x)$. Тогда $\{y, x_1, \ldots, x_n\}$ — множество свободных переменных формулы $(\forall x)\, A(x) \to A(y)$. Пусть $\mathfrak{M}$ — произвольная модель (сигнатура которой содержит сигнатуру формулы $A(x)$).

Рассмотрим произвольный набор $(\beta_0, \alpha_1, \ldots, \alpha_n)$ значений переменных $(y, x_1, \ldots, x_n)$ в модели $\mathfrak{M}$. Если для некоторого набора $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ значений переменных $(x, x_1, \ldots, x_n)$

$$A(x)|_{(\alpha_0, \alpha_1, \ldots, \alpha_n)} = 0,$$

то $(\forall x)A(x)|_{(\alpha_1, \ldots, \alpha_n)} = 0$ и

$$((\forall x)\, A(x) \to A(y))|_{(\beta_0, \alpha_1, \ldots, \alpha_n)} = 1.$$

Если же для всех наборов $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ (где $\alpha_1, \ldots, \alpha_n$ фиксированы, а α_0 — произвольный элемент)

$$A(x)|_{(\alpha_0, \alpha_1, \ldots, \alpha_n)} = 1,$$

то $A(y)|_{(\beta_0, \alpha_1, \ldots, \alpha_n)} = 1$. Поэтому

$$((\forall x)\, A(x) \to A(y))|_{(\beta_0, \alpha_1, \ldots, \alpha_n)} = 1.$$

Схема аксиом (5). Рассмотрим формулу

$$A(x) \to (\exists y)\, A(y).$$

Пусть $\{x, x_1, \ldots, x_n\}$ — множество свободных переменных формулы $A(x)$. Тогда формула $A(x) \to (\exists y)\, A(y)$ имеет то же множество $\{x, x_1, \ldots, x_n\}$ свободных переменных. Пусть $\mathfrak{M}$ — произвольная модель, сигнатура которой содержит сигнатуру формулы $A(x)$. Рассмотрим произвольный набор $\tilde{\alpha} = (\alpha_0, \alpha_1, \ldots, \alpha_n)$ значений переменных $(x, x_1, \ldots, x_n)$.

Если $A(x)|_{\tilde{\alpha}} = 0$, то

$$(A(x) \to (\exists y)\, A(y))|_{\tilde{\alpha}} = 1.$$

Если же $A(x)|_{\tilde{\alpha}} = 1$, то $(\exists y)\, A(y)|_{(\alpha_1, \alpha_2, \ldots, \alpha_n)} = 1$. Поэтому

$$(A(x) \to (\exists y)\, A(y))|_{\tilde{\alpha}} = 1.$$

Лемма 2. *Формула, полученная из тождественно истинных формул по любому из правил вывода* I–V*, является тождественно истинной.*

Доказательство. Для правила вывода I это утверждение было установлено при доказательстве тождественной истинности формул исчисления высказываний.

Рассмотрим правило вывода II:

$$\frac{B \to A(x)}{B \to (\forall x)\, A(x)}.$$

Пусть $\{x_1, \ldots, x_n\}$ — множество свободных переменных формулы $B \to (\forall x)\, A(x)$. Тогда $\{x, x_1, \ldots, x_n\}$ — множество свободных переменных формулы $B \to A(x)$. Пусть $\mathfrak{M}$ — произвольная модель.

Рассмотрим произвольный набор $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ значений переменных $(x_1, \ldots, x_n)$ в модели $\mathfrak{M}$.

Если $B|_{\tilde{\alpha}} = 0$, то

$$(B \to (\forall x)\, A(x))|_{\tilde{\alpha}} = 1.$$

Если же $B|_{\tilde{\alpha}} = 1$, то поскольку по условию леммы формула $B \to A(x)$ является тождественно истинной, то для любого набора $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ значений переменных $(x, x_1, \ldots, x_n)$

$$(B \to A(x))|_{(\alpha_0, \alpha_1, \ldots, \alpha_n)} = 1.$$

Поэтому (так как $B|_{\tilde{\alpha}} = 1$)

$$A(x)|_{(\alpha_0, \alpha_1, \ldots, \alpha_n)} = 1.$$

Следовательно,

$$(\forall x)\, A(x)|_{\tilde{\alpha}} = 1.$$

Рассмотрим правило вывода III:

$$\frac{A(x) \to B}{(\exists x)\, A(x) \to B}.$$

Пусть $\{x_1, \ldots, x_n\}$ — множество свободных переменных формулы $(\exists x)\, A(x) \to B$. Тогда $\{x, x_1, \ldots, x_n\}$ — множество свободных переменных формулы $A(x) \to B$. Пусть $\mathfrak{M}$ — произвольная модель. Рассмотрим произвольный набор $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ значений переменных $(x_1, \ldots, x_n)$ в модели $\mathfrak{M}$.

Если $B|_{\tilde{\alpha}} = 1$, то

$$((\exists x)\, A(x) \to B)|_{\tilde{\alpha}} = 1.$$

Если же $B|_{\tilde{\alpha}} = 0$, то поскольку по условию леммы формула $A(x) \to B$ является тождественно истинной, то для любого набора $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ значений переменных $(x, x_1, \ldots, x_n)$

$$(A(x) \to B)|_{(\alpha_0, \alpha_1, \ldots, \alpha_n)} = 1.$$

Поэтому

$$A(x)|_{(\alpha_0, \alpha_1, \ldots, \alpha_n)} = 0.$$

Следовательно,

$$(\forall x)\, A(x)|_{\tilde{\alpha}} = 0$$

и

$$((\exists x)\, A(x) \to B)|_{\tilde{\alpha}} = 1.$$

Справедливость леммы для правил IV и V очевидна.

Теорема 2. *Любая выводимая формула в исчислении предикатов является тождественно истинной.*

Утверждение теоремы непосредственно следует из лемм 1 и 2.

Справедлива и обратная[1)]

Теорема (Гёделя о полноте**).** *Любая тождественно истинная формула является выводимой.*

Доказательство этого утверждения здесь приводиться не будет.

Из теоремы 2 непосредственно следует

Теорема 3. *Исчисление предикатов непротиворечиво.*

Докажем выводимость некоторых формул.

В рассматриваемых ниже примерах $A(x, y)$ — произвольная формула, у которой x и y — свободные переменные.

Примеры.

1. $\vdash (\forall x)(\forall y)\, A(x, y) \to (\forall y)(\forall x)\, A(x, y)$.

Доказательство.

1. $(\forall x)(\forall y)\, A(x, y) \to (\forall y)\, A(z, y)$ — схема аксиом (4)
2. $(\forall y)A(z, y) \to A(z, v)$ — схема аксиом (4)

Из ослабленной теоремы о дедукции для любых формул A, B и C следует, что

$$A \to B,\ B \to C \vdash_C A \to C.$$

Поэтому в силу 1, 2

$$\vdash (\forall x)(\forall y)\, A(x, y) \to A(z, v).$$

По правилу вывода II из полученной формулы будем иметь

$$\vdash (\forall x)(\forall y)\, A(x, y) \to (\forall z)\, A(z, v),$$

откуда в соответствии с правилом вывода II получаем

$$\vdash (\forall x)(\forall y) A(x, y) \to (\forall v)(\forall z) A(z, v).$$

[1)] Доказательство см., например, в книге: *Новиков П. С.* Элементы математической логики. М.: Наука, 1973.

Применяя к этой формуле правило вывода V замены связанной переменной, получаем сначала

$$\vdash (\forall x)(\forall y)\, A(x,y) \to (\forall y)(\forall z)\, A(z,y),$$

а затем (по правилу вывода V)

$$\vdash (\forall x)(\forall y) A(x,y) \to (\forall y)(\forall x) A(x,y).$$

2. $\vdash (\exists x)(\exists y)\, A(x,y) \to (\exists y)(\exists x)\, A(x,y)$.

Доказательство.

1. $A(z,v) \to (\exists x)\, A(x,v)$ — схема аксиом (5)
2. $(\exists x)\, A(x,v) \to (\exists y)(\exists x)\, A(x,y)$ — схема аксиом (5)

Так как для любых формул A, B и C

$$A \to B, B \to C \vdash_C A \to C,$$

то в силу 1, 2 имеем

$$\vdash A(z,v) \to (\exists y)(\exists x) A(x,y).$$

По правилу вывода III из полученной формулы получаем

$$\vdash (\exists v)\, A(z,v) \to (\exists y)(\exists x)\, A(x,y),$$

откуда по правилу вывода III будем иметь

$$\vdash (\exists z)(\exists v)\, A(z,v) \to (\exists y)(\exists x)\, A(x,y).$$

Применяя к этой формуле два раза правило вывода V, получаем сначала

$$\vdash (\exists x)(\exists v)\, A(x,v) \to (\exists y)(\exists x)\, A(x,y),$$

а затем

$$\vdash (\exists x)(\exists y)\, A(x,y) \to (\exists y)(\exists x)\, A(x,y).$$

3. $\vdash (\exists x)(\forall y)\, A(x,y) \to (\forall y)(\exists x)\, A(x,y)$.

Доказательство.

1. $(\forall y)\, A(x,y) \to A(x,v)$ — схема аксиом (4)
2. $A(x,v) \to (\exists z)\, A(z,v)$ — схема аксиом (5)

Так как для любых формул A, B и C

$$A \to B, B \to C \vdash_C A \to C,$$

то в силу 1, 2

$$\vdash (\forall y)\, A(x,y) \to (\exists z)\, A(z,v).$$

По правилу вывода III из полученной формулы будем иметь

$$\vdash (\exists x)(\forall y)\, A(x,y) \to (\exists z)\, A(z,v),$$

откуда по правилу вывода II получаем

$$\vdash (\exists x)(\forall y)\, A(x,y) \to (\forall v)(\exists z)\, A(z,v).$$

Применяя к этой формуле правило вывода V, получаем

$$\vdash (\exists x)(\forall y)\, A(x,y) \to (\forall y)(\exists z)\, A(z,y),$$

и, наконец, по правилу вывода V

$$\vdash (\exists x)(\forall y)\, A(x,y) \to (\forall y)(\exists x)\, A(x,y).$$

Отметим, что формула

$$(\forall x)(\exists y)\, A(x,y) \to (\exists y)(\forall x)\, A(x,y)$$

не является выводимой, поскольку не является тождественно истинной. В самом деле, в модели $\mathfrak{M} = \langle M, \Sigma\rangle$, где $M = \{a,b\}$, $\Sigma = \{A^{(2)}\}$, а $A^{(2)}(x,y)$ — предикат равенства (т. е. $A^{(2)}(a,a) = A^{(2)}(b,b) = 1$, $A^{(2)}(a,b) = A^{(2)}(b,a) = 0$), приведенная выше формула является ложной.

Учебное пособие

Конспект лекций О. Б. Лупанова по курсу "Введение в математическую логику"

Ответственный редактор: Александр Борисович Угольников

Редактор: Н. А. Леонтьева

Компьютерный набор: П. А. Бородин, Ю. В. Бородина

Оригинал-макет: В. М. Староверов, О. С. Дудакова

Подписано в печать 04.06.2007 г.
Формат 60×90 1/16. Усл. печ. л. 12
Заказ Тираж 500 экз.

Издательство Центра прикладных исследований при механико-математическом факультете МГУ
г. Москва, Ленинские горы.
Изд. лиц. № 04059 от 20.02.2001 г.

Отпечатано на типографском оборудовании механико-математического факультета и Франко-русского центра им. А. М. Ляпунова МГУ