Министерство образования и науки РФ

Федеральное государственное бюджетное
образовательное учреждение
высшего профессионального образования
«Тверской государственный университет»

М.И. Дехтярь, Б.Н. Карлов

ЗАДАЧНИК ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ

Учебное пособие по курсу «Дискретная математика» для направлений «Фундаментальная информатика и информационные технологии», «Прикладная математика и информатика», «Бизнес-информатика» и «Прикладная информатика».

Тверь — 2013

УДК 510.5 + 519.1 (075.8)
ББК В174я73-4
Д 39

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Тверской государственный университет»
Факультет прикладной математики и кибернетики

М.И. Дехтярь, Б.Н. Карлов. Задачник по дискретной математике. Учебное пособие / Тверь, Тверской государственный университет, 2013 г.

Настоящий сборник задач является пособием для практических занятий по дискретной математике и предназначен в основном для студентов первого курса Тверского государственного университета, которые обучаются по направлениям 010300.62 — «Фундаментальная информатика и информационные технологии», 010400.62 —«Прикладная математика и информатика», 080500.62 —«Бизнес-информатика» и 230700.62 — «Прикладная информатика».

Он может быть использован преподавателями и студентами для подготовки к практическим занятиям и контрольным работам.

В книге помещено около 400 задач и упражнений, многие из которых состоят из нескольких независимых подзадач. Большинство задач снабжено ответами, а многие более сложные задачи сопровождаются также указаниями или решениями.

Задачи в сборнике собраны в 23 раздела. Они посвящены элементам теории множеств и комбинаторики, булевым функциям и их реализациям, логике предикатов, конечным графам, конечным автоматам и началам теории алгоритмов. Каждому разделу предшествует небольшое теоретическое введение, содержащее основные определения и формулы, используемые при решении его задач.

УДК 510.5 + 519.1 (075.8)
ББК В174я73-4

Оглавление

Предисловие

Настоящий сборник задач является пособием для практических занятий по дискретной математике и предназначен в основном для студентов первого курса Тверского государственного университета, которые обучаются по направлениям 010300.62 — «Фундаментальная информатика и информационные технологии», 010400.62 —«Прикладная математика и информатика», 080500.62 —«Бизнес-информатика» и 230700.62 —«Прикладная информатика». Он может быть использован преподавателями для подготовки к практическим занятиям и контрольным работам.

По темам задач и по структуре, а также по терминологии и обозначениям данный задачник является дополнением к учебному пособию [4], которое на протяжении ряда лет используется в ТвГУ в качестве основного учебника по курсу дискретной математики. В частности, в задачник вошли почти все задачи из [4]. Опыт преподавания показал, что этот набор следовало бы существенно расширить. Другие известные нам пособия и задачники[1] по дискретной математике не содержат всех существенных для нашей программы разделов. Например, во многих отсутствуют основы теории конечных автоматов и алгоритмов, нигде нет задач по Хорновским формулам и упорядоченным бинарным диаграммам решений.

Задачник может использоваться и как независимое пособие, поскольку задачам каждого раздела предшествует небольшое теоретическое введение, содержащее основные определения и формулы, используемые при их решении.

В книге помещено около 400 задач и упражнений, многие из которых состоят из нескольких независимых подзадач. Условно их можно отнести к следующим категориям.

(1) Упражнения на закрепление основного материала, в частности, задачи на выполнение стандартных преобразований и на “прокрутку” тех или иных процедур и алгоритмов.

(2) Задачи на доказательство (как правило, не очень сложное) отдельных утверждений об основных изучаемых понятиях.

(3) Задачи на установление свойств новых, не определённых в основном тексте понятий. Эти задачи расширяют материал книги [4] и предназначены для самых любознательных читателей.

Большинство задач типа (1) снабжено ответами, а многие задачи типов

[1] Некоторые из них, послужившие источниками ряда задач, указаны в списке литературы.

(2) и (3) сопровождаются также указаниями и решениями.

Задачи в сборнике собраны в 23 раздела. Первые три являются введением и позволяют получить первоначальные представления о множествах, отношениях и функциях, приобрести некоторые навыки в проведении доказательств методом математической индукции, научиться применять основные формулы комбинаторики. Разделы 4–9 посвящены булевым функциям. Тематика первых пяти из них достаточно традиционна. Задачи раздела 9 посвящены интересному подклассу булевых формул — хорновским формулам — и эффективным алгоритмам проверки выводимости для них. Задачи разделов 10 и 11 предназначены для студентов специальностей, по которым не предусмотрены отдельные курсы по математической логике. Они в основном должны научить пониманию семантики формул логики предикатов и продемонстрировать её связь с теорией баз данных. Разделы 12–14 включают задачи, связанные с основами теории графов, важным подклассом графов – деревьями, а также несколькими классическими алгоритмами на графах. Разделы 15 и 16 посвящены двум разным способам представления булевых функций с помощью графов: логическим схемам и упорядоченным бинарным диаграммам решений. Конечные автоматы рассматриваются в разделах 17–19. Учащиеся должны научиться строить конечные автоматы и регулярные выражения для достаточно простых языков, детерминизировать недетерминированные конечные автоматы и использовать лемму о разрастании для доказательства неавтоматности некоторых языков. Последняя часть задачника посвящена началам теории алгоритмов. Основные задачи в разделах 20–22 относятся к реализации различных арифметических функций с помощью одного из трёх алгоритмических языков: структурированных программ, частично рекурсивных определений и машин Тьюринга. В задачах раздела 23 содержится материал о моделировании одних моделей алгоритмов другими и о неразрешимых алгоритмических проблемах.

Электронные адреса авторов: Michael.Dekhtyar@tversu.ru и bnkarlov@gmail.com. Мы будем искренне благодарны за замечания и советы.

1. Множества, отношения и функции

Операции над множествами.
Объединение множеств: $A \cup B = \{x \mid x \in A \text{ } \textit{или} \text{ } x \in B\}$.
Пересечение множеств: $A \cap B = \{x \mid x \in A \text{ } \textit{и} \text{ } x \in B\}$.
Разность множеств: $A \setminus B = \{x \mid x \in A \text{ } \textit{и} \text{ } x \notin B\}$.
Дополнение множества A (в универсуме U): $\bar{A} = U \setminus A$.
Симметрическая разность (дизъюнктивная сумма): $A \dot{-} B = (A \setminus B) \cup (B \setminus A)$.
Декартово произведение множеств $A_1, \ldots, A_n$:
$A_1 \times \ldots \times A_n = \{(a_1, \ldots, a_n) \mid a_1 \in A_1, \ldots, a_n \in A_n\}$.
$\varnothing$ — пустое множество: для любого x $x \notin \varnothing$.
Декартова степень: При $n \geq 1$ $A^n = A \times A \times \ldots \times A$ (n раз).
$\mathbf{N} = \{0, 1, 2, 3, \ldots\}$ — множество натуральных чисел.
$\mathbf{R}$ — множество всех вещественных (действительных) чисел.

Отношения и функции.
Бинарное или двуместное отношение между элементами множеств A и B — подмножество R их декартова произведения $A \times B$.
Область определения R: $\delta_R = \{x \mid \text{существует } y \text{ такое, что } (x, y) \in R\}$.
Область значений R: $\rho_R = \{y \mid \text{существует } x \text{ такое, что } (x, y) \in R\}$.
Обратное отношение для бинарного отношения R: $R^{-1} = \{(x, y) \mid (y, x) \in R\}$.
Произведение отношений $R_1 \subseteq A \times B$ и $R_2 \subseteq B \times C$:
$R_1 \circ R_2 = \{(x, z) \mid \text{ существует } y \in B \text{ такой, что } (x, y) \in R_1 \text{ и } (y, z) \in R_2\}$.
Бинарное отношение R на множестве A называется *отношением эквивалентности*, если для него выполнены следующие условия:
1) *рефлексивность*: для любого $a \in A$ $(a, a) \in R$;
2) *симметричность*: для любых a, b из A $(a, b) \in R \Leftrightarrow (b, a) \in R$;
3) *транзитивность*: для любых трёх элементов a, b, c из A, если $(a, b) \in R$ и $(b, c) \in R$, то и $(a, c) \in R$.
Классы эквивалентности для R : для $a \in A$ его класс эквивалентности $[a]_\equiv$ включает все эквивалентные a элементы: $[a]_R = \{b \in A \mid R(a, b)\}$.
Отношение R на множестве A называется *отношением частичного порядка*, если для него выполнены следующие условия:
1) *антирефлексивность*: для любого $a \in A$ $(a, a) \notin R$;
2) *антисимметричность*: для любых a, b из A, если $(a, b) \in R$ и $(b, a) \in R$, то $a = b$;
3) *транзитивность*: для любых трёх элементов a, b, c из A, если $(a, b) \in R$ и $(b, c) \in R$, то и $(a, c) \in R$.
R — отношение линейного порядка, если для него выполнены условия 1–3 и
4) *Линейность*: для любых a, b из A либо $(a, b) \in R$, либо $(b, a) \in R$.

Отношение f называется *функцией*, или *отображением*, из A в B (из A *на* B) , если $\delta_f = A$, $\rho_f \subseteq B$ (соответственно, $\rho_f = B$) и для всех x, y_1, y_2 из того, что $(x, y_1) \in f$ и $(x, y_2) \in f$, следует, что $y_1 = y_2$.
Запись: $f : A \to B$. Для функции f вместо $(x, y) \in f$ пишем $f(x) = y$.
Если $\delta_f = A \subsetneq A'$, то f называется *частичной функцией* из A' в B.
f называется *1-1-функцией* (или *обратимой функцией*), если для любых x_1, x_2, y из того, что $f(x_1) = y$ и $f(x_2) = y$ следует, что $x_1 = x_2$.
Функция $f : A \to B$ называется *взаимно однозначной функцией,* если она является 1-1-функцией и $\rho_f = B$. Взаимно однозначная функция $f : A \to A$ называется *перестановкой множества* A.
Иногда функции *на* называют *сюръективными*, а *1-1-функции* — *инъективными.*
Образ множества $C \subseteq A$ при отображении $f : A \to B$ — это множество $f(C) = \{d \mid f(c) = d \text{ для некоторого } c \in C\}$.
Прообраз множества $D \subseteq B$ при отображении $f : A \to B$ — это множество $f^{-1}(D) = \{c \mid f(c) = d \text{ для некоторого } d \in D\}$.

Множества A и B называются *равномощными*, если существует взаимно однозначная функция $f : A \to B$.
Множество называется *счётным*, если оно равномощно множеству натуральных чисел $\mathbf{N}$.

Задачи

1.1. Найти все подмножества множеств $\varnothing$, $\{\varnothing\}$, $\{1, 2, 3\}$, $\{a, \{1, 2\}, \varnothing\}$.

1.2. Пусть множество $A = \{0, \{0, 1, 2\}, \{3\}, 4, \{\{5\}\}, 6\}$. Какие из следующих множеств $B = \{0, 4\}, C = \{6, \{3\}, 0\}, D = \{0, 3\}, E = \{\{0, 1, 2\}, \{3\}\}, F = \{0, \{5\}\}, G = \{\{3\}, 2, \{\{5\}\}, 6\}$ не являются подмножествами множества A?

1.3. Пусть заданы три множества: $A = \{a, b, \{\varnothing\}, \{a, c, d\}\}$, $B = \{a, c, e, \{a\}, \{b\}\}$, $C = \{a, b, c, d, \{e\}, \varnothing\}$. Найдите множество $D = (A \cup B) \setminus C$. Какова его мощность?

1.4. Пусть заданы три множества: $A = \{a, b, c, \{\varnothing\}, \{a\}\}$, $B = \{a, e, \{a\}, \{b\}, \varnothing\}$, $C = \{a, b, d, \{e\}, \{\varnothing\}\}$. Найдите множество $D = (A \setminus B) \cap C$. Какова его мощность?

1.5. Пусть $A = \{0, 1\}$, $B = \{a, b, c\}$. Определите множества $A \times B$ и $B \times A$.

1.6. Для каких множеств A и B выполняется равенство $A \times B = B \times A$?

1.7. Пусть заданы множества $A = \{0, 1, 2\}, B = \{2, 3\}, C = \{a, b, c\}$ и $D = \{a, c, e\}$. Определите, чему равны следующие множества:
а) $F_1 = (A \setminus B) \times (C \cap D)$;
б) $F_2 = (A \cap B) \times (C \cap D)$;
в) $F_3 = (B \setminus A) \times (C \setminus D)$.

1.8. Доказать следующие включения:
а) $A \cap B \subseteq A \subseteq A \cup B$;
б) $A \setminus B \subseteq A$.

1.9. Доказать следующие тождества для любых множеств A, B, C:
а) $A \cup A = A \cap A = A$;
б) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$;
в) $(A \cup B) \cap A = (A \cap B) \cup A = A$;
г) $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$;
д) $A \setminus (B \setminus C) = (A \setminus B) \cup (A \cap C)$;
е) $A \cup \varnothing = \varnothing \cup A = A$;
ж) $A \cap \varnothing = \varnothing \cap A = \varnothing$;
з) $A \dot{-} \varnothing = \varnothing \dot{-} A = A$;
и) $A \dot{-} A = \varnothing$.

1.10. Пусть множества A, B, C и их дополнения являются подмножествами универсума U. Доказать следующие тождества:
а) $\overline{A \cap B} = \bar{A} \cup \bar{B}$;
б) $\overline{A \cup B} = \bar{A} \cap \bar{B}$;
в) $A \setminus \bar{B} = A \cap B$;
г) $\bar{A} \setminus B = \bar{A} \cap \bar{B}$;
д) $\bar{A} \setminus \bar{B} = B \setminus A$.

1.11. Доказать, что
а) $A \times (B \cup C) = (A \times B) \cup (A \times C)$;
б) $A \times (B \cap C) = (A \times B) \cap (A \times C)$;
в) $A \times (B \setminus C) = (A \times B) \setminus (A \times C)$;
г) если $A \subseteq B$ и $C \subseteq D$, то $(A \times C) = (A \times D) \cap (B \times C)$.

1.12. Для каждого из следующих отношений определить $\delta_R, \rho_R, R^{-1}, R \circ R, R \circ R^{-1}$:
а) $R = \{(x, y) \mid x, y \in \mathbf{N}$ и x делит $y\}$;
б) $R = \{(x, y) \mid x, y \in \mathbf{N}$ и $x + y \leq 10\}$;
в) $R = \{(x, y) \mid x, y \in \mathbf{N}$ и $y = 3x + 1\}$;

г) $R = \{(x, x^2) \mid x \in \mathbf{N}$ и $x \leq 10\}$;
д) $R = \{(a,b), (b,c), (b,d), (c,d), (d,b)\}$.

1.13. Для каждого из следующих бинарных отношений на множестве $A = \{a, b, c\}$ определите, является ли оно рефлексивным, симметричным, антисимметричным, транзитивным:
а) $R_1 = \{(a,a), (a,b), (b,a), (b,b), (c,c)\}$;
б) $R_2 = \{(a,a), (a,c), (c,b), (a,b), (b,b), (c,c)\}$;
в) $R_3 = \{(a,a), (a,c), (c,b), (a,b)\}$.

1.14. На множестве всех непустых отрезков числовой прямой определены три отношения: $P = \{([a,b],[c,d]) | c < a < b < d\}, Q = \{([a,b],[c,d]) | a < c < b < d\}$ и $R = \{([a,b],[c,d]) | b < c\}$. Определите, какие из них являются отношениями частичного порядка.

1.15. Пусть бинарные отношения P и Q на множестве A являются рефлексивными. Определите, какие из следующих отношений также являются рефлексивными:
а) $P \cap Q$;
б) $P \cup Q$;
в) $P \circ Q$;
г) P^{-1}.

1.16. Пусть бинарные отношения P и Q на множестве A являются симметричными Определите, какие из следующих отношений также являются симметричными:
а) $P \cap Q$;
б) $P \cup Q$;
в) $P \circ Q$;
г) P^{-1}.

1.17. Пусть бинарные отношения P и Q на множестве A являются антисимметричными Определите, какие из следующих отношений также являются антисимметричными:
а) $P \cap Q$;
б) $P \cup Q$;
в) $P \circ Q$;
г) P^{-1}.

1.18. Пусть бинарные отношения P и Q на множестве A являются транзитивными. Определите, какие из следующих отношений также являются

транзитивными:
а) $P \cap Q$;
б) $P \cup Q$;
в) $P \circ Q$;
г) P^{-1}.

1.19. Пусть множество $S = \{(i, j) \mid 1 \leq i, j \leq 8\}$ задает клетки шахматной доски. Опишите следующие бинарные отношения на S:
а) $L = \{(a, b) \mid$ ладья за 1 ход может перейти с клетки a на клетку $b\}$;
б) $K = \{(a, b) \mid$ конь за 1 ход может перейти с клетки a на клетку $b\}$.
Будут ли эти отношения эквивалентностями?
Опишите отношение $L \circ L$.

1.20. Пусть $A = \{1, 2, 3, 4, 5\}$. Определите, какие из следующих отношений на A являются отношениями эквивалентности. Для тех из них, которые являются отношениями эквивалентности, перечислите их классы эквивалентности.
а) $\{(1,1), (2,2), (3,3), (4,4), (5,5), (1,4), (4,1)\}$;
б) $\{(1,2), (2,1), (3,3), (4,5), (5,4), (5,5)\}$;
в) $\{(1,1), (2,2), (3,3), (4,4), (5,5), (1,4), (4,1), (4,2), (2,4)\}$;
г) $\{(1,1), (2,2), (3,3), (4,4), (5,5), (1,2), (2,1), (2,4), (4,2), (3,5), (5,3), (1,4), (4,1)\}$;
д) $\{(a, b) \mid a \in A, b \in A$ и $a + b$ делится на $3\}$;
е) $\{(a, b) \mid a \in A, b \in A$ и $a - b$ делится на $2\}$;
ж) $\{(a, b) \mid a \in A, b \in A$ и $a + 3$ делится на $b\}$.

1.21. Пусть Π — множество прямых на плоскости. Будут ли следующие отношения на Π отношениями эквивалентности:
а) параллельность прямых (будем считать, что прямая параллельна себе самой);
б) перпендикулярность прямых?

1.22. Для каждого из следующих бинарных отношений над множеством натуральных чисел $\mathbf{N}$ определите, является ли оно рефлексивным, антирефлексивным, симметричным, антисимметричным, транзитивным:
а) $D_1 = \{(x, y) \mid x + 1$ делится на $y + 1\}$;
б) $D_2 = \{(x, y) \mid x + 1$ делится на $y + 1$ или $y + 1$ делится на $x + 1\}$;
в) $D_3 = \{(x, y) \mid x + 1$ не делится на $y + 1\}$.

1.23. Для каждого из следующих трёх отношений R_i $(i = 1, 2, 3)$, определённых на совокупности всех непустых подмножеств действительных (ве-

щественных) чисел, определите являются ли они рефлексивными, симметричными, антисимметричными, транзитивными, отношениями частичного порядка.
а) $R_1 = \{(A, B) \mid$ для любого $\varepsilon > 0$ существуют $a \in A$ и $b \in B$ такие, что $|a - b| \leq \varepsilon\}$,
б) $R_2 = \{(A, B) \mid$ для любых $a \in A$ и $\varepsilon > 0$ существует $b \in B$ такое, что $|a - b| \leq \varepsilon\}$,
в) $R_3 = \{(A, B) \mid$ для любых $a \in A, b \in B$ и $\varepsilon > 0$ существуют $a' \in A$ и $b' \in B$ такие, что $|a - b'| \leq \varepsilon$ и $|a' - b| \leq \varepsilon\}$.

1.24. Пусть $A = \{a_1, \ldots, a_m\}$ — произвольный конечный алфавит. Обозначим через A^n множество слов длины n в алфавите A (это обозначение согласовано с тем же обозначением декартовой степени A, так как степень A^n состоит из всех последовательностей элементов A длины n). Через A^* обозначим множество всех слов в алфавите A.
а) Определим следующее отношение R_1 на словах из A^n.
Пусть $v = a_{i_1} a_{i_2} \ldots a_{i_n}$, $w = a_{j_1} a_{j_2} \ldots a_{j_n}$. Тогда
$(v, w) \in R_1$ тогда и только тогда, когда для всех k от 1 до n $i_k \leq j_k$ и для некоторого такого k $i_k < j_k$, т.е. номер каждой буквы слова v не больше номера той же буквы в слове w и хотя бы у одной из букв он меньше.
Является ли это отношение R_1 отношением частичного (линейного) порядка?
б) Определим следующее отношение R_2 на словах из A^*.
Пусть $v = a_{i_1} a_{i_2} \ldots a_{i_n}$, $w = a_{j_1} a_{j_2} \ldots a_{j_r}$. Тогда
$(v, w) \in R_2$ тогда и только тогда, когда существует такое k в интервале от 1 до n, что при $l < k$ $i_l = j_l$ и $i_k < j_k$ или $n < r$ и первые n символов w совпадают со словом v.
Является ли это отношение R_2 отношением частичного (линейного) порядка?

Замечание. Определенное в пункте (а) отношение R_1 называется *отношением покоординатного порядка,* а отношение R_2 из пункта (б) — *отношением лексикографического порядка.* В соответствии с лексикографическим порядком упорядочены, например, слова в словарях и энциклопедиях.

1.25. Пусть $\mathbf{R}$ — это множество всех вещественных чисел. Для каждой из следующих функций найдите её область значений и определите, какие из них являются 1-1- или взаимно однозначными функциями.
а) $f : \mathbf{R} \to \mathbf{R}$, $f(x) = 3x + 1$;

б) $f : \mathbf{R} \to \mathbf{R},\ f(x) = x^2 + 1$;
в) $f : \mathbf{R} \to \mathbf{R},\ f(x) = x^3 - 1$;
г) $f : \mathbf{R} \to \mathbf{R},\ f(x) = e^x$;
д) $f : \mathbf{R} \to \mathbf{R},\ f(x) = \sqrt{3x^2 + 1}$;
е) $f : [-\pi/2, \pi/2] \to \mathbf{R},\ f(x) = \sin x$;
ж) $f : [0, \pi] \to \mathbf{R},\ f(x) = \sin x$.

1.26. Пусть $g : A \to B$ и $f : B \to C$ — две функции. Пусть $f \circ g : A \to C$ — это суперпозиция этих функций, заданная соотношением $(f \circ g)(x) = f(g(x))$. Определите, какие из следующих утверждений верны.
а) Если g является 1-1-функцией, то $f \circ g$ также является 1-1-функцией.
б) Если g и f являются функциями "на", т.е. $\rho_g = B, \rho_f = C$, то $f \circ g$ также является функцией "на ", т.е. $\rho_{f \circ g} = C$.
в) Если g и f являются взаимно однозначными функциями, то $f \circ g$ также является взаимно однозначной функцией.
г) Если $f \circ g$ является 1-1-функцией, то f также является 1-1-функцией.
д) Если $f \circ g$ является 1-1-функцией, то g также является 1-1-функцией.
е) Если $f \circ g$ является функцией "на", т.е. $\rho_{f \circ g} = C$, то f также является функцией "на", т.е. $\rho_f = C$.

1.27. Доказать, что если множества A и B конечны, то
а) $|A \times B| = |A| \cdot |B|$;
б) $|A \cup B| = |A| + |B| - |A \cap B|$;
в) $|\{0, 1\}^n| = 2^n$.

1.28. Докажите, что следующие множества счётны:
а) множество всех точек плоскости с целочисленными координатами;
б) множество всех векторов размера $n > 0$ с неотрицательными целочисленными координатами, т.е. множество $\mathbf{N}^n = \{(k_1, k_2, \ldots, k_n) \mid k_i \in \mathbf{N}, i = 1, 2, \ldots, n\}$;
в) множество всех векторов с неотрицательными целочисленными координатами, т.е. множество $\cup_{n=1}^{\infty} \mathbf{N}^n$;
г) множество всех слов в алфавите с m символами;
д) множество всех рациональных чисел;
е) множество всех многочленов с целыми коэффициентами.

2. Метод математической индукции

Пусть $P(n)$ это некоторое утверждение, однозначно зависящее от целочисленного параметра n.
Доказательство утверждения “для всех $n \geq n_0$ верно $P(n)$” *по индукции* включает два этапа.
1) *Базис индукции* состоит в доказательстве утверждения $P(n_0)$ для некоторого начального значения n_0.
2) *Шаг индукции* состоит в предположении справедливости $P(n)$ при $n = k \geq n_0$ и доказательстве из этого предположения справедливости утверждения $P(k+1)$.

Задачи

2.1. Используя индукцию, доказать, что
а) $2n+1 \leq 2^n$ при $n \geq 3$;
б) $n^2 \leq 2^n$ при $n \geq 4$.

2.2. Используя индукцию, доказать, что при $n \geq 1$
а) $(7^n - 1)$ делится нацело на 6;
б) $(\ 11^n - 6)$ делится нацело на 5;
в) $(67^n - 23^n)$ делится нацело на 4;
г) $(3^n + 7^n - 2)$ делится нацело на 8.

2.3. Доказать, что $1^4 + 2^4 + \ldots + n^4 = \frac{1}{30}n(n+1)(2n+1)(3n^2+3n-1)$.

2.4. Доказать, что $1 \cdot 2 + 2 \cdot 3 + \ldots + n(n+1) = \frac{n(n+1)(n+2)}{3}$.

2.5. Доказать, что $1 + x + x^2 + x^3 + \ldots + x^n = \frac{x^{n+1}-1}{x-1}$.

2.6. Доказать, что n различных прямых на плоскости разбивают её на области, которые можно закрасить белой и чёрной красками так, что смежные области будут закрашены разными красками.

2.7. Доказать, что n различных прямых на плоскости, никакие две из которых не параллельны и никакие три из которых не пересекаются в одной точке, разбивают плоскость на $(n^2+n+2)/2$ областей.

2.8. Докажите, что если стоимость некоторого письма ≥ 6 руб., то его можно точно оплатить, используя 2- и 7-рублёвые марки.

2.9. Найдите ошибку в следующем доказательстве “по индукции” утверждения:
для всех $n \geq 1$ справедливо неравенство $3^n > 3(n+1)+1$.

Пусть для некоторого $k \geq 1$ неравенство справедливо, т.е. $3^k > 3(k+1)+1$ $(*)$. Докажем, что оно верно и для $n = k+1$, т.е. $3^{k+1} > 3(k+2)+1$. Для этого заметим, что для любого $k \geq 1$ верно неравенство $2 \cdot 3^k > 3$. Прибавив его левую и правую часть к соответствующим частям неравенства $(*)$, получим $3^k + 2 \cdot 3^k > 3(k+1)+1+3$ или $3^{k+1} > 3(k+2)+1$, что и требовалось.
Установите, при каких n справедливо неравенство $3^n > 3(n+1)+1$.

2.10. Предположим, что некоторая колония бактерий $\mathcal{C}$ состоит из конечного числа бактерий, для каждой из которых указано натуральное число – предполагаемый срок жизни, т.е. $\mathcal{C} = \{< b, t >|$ b — идентификатор бактерии, t — срок её жизни$\}$. На каждом шаге в жизни колонии происходит одно событие: выбирается какая-нибудь бактерия $< b, t >$ и, если $t = 0$, то она удаляется из колонии (“умирает”), а если $t > 0$, то эта бактерия заменяется на некоторое конечное множество бактерий, срок жизни каждой из которых меньше t (“размножается”).

Докажите, что любая такая колония $\mathcal{C}$ в конце концов станет пустой (“вымрет”).

Указание: используйте индукцию по времени жизни старейшей бактерии T: $T = \max\{t \mid\ < b, t > \in \mathcal{C}\}$.

2.11. Числа Фибоначчи определяются следующими реккурентными соотношениями:
$F_0 = 0$, $F_1 = 1$, $F_{n+2} = F_{n+1} + F_n$. Докажите, что
а)

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right];$$

б)

$$F_1 + F_2 + F_3 + \ldots + F_n = F_{n+2} - 1.$$

2.12. (Ханойские башни.) Имеется три стержня. На один из них надето несколько дисков, диаметры которых убывают снизу вверх (получается детская пирамидка). Разрешается перекладывать верхний диск с любого стержня на любой другой, но при этом запрещено класть диск большего диаметра на диск меньшего диаметра. Требуется перенести всю пирамидку на третий стержень.

а) Докажите, что n дисков можно переместить за $2^n - 1$ операций.
б) Докажите, что n дисков нельзя переместить меньше чем за $2^n - 1$, операций.
в) Легенда гласит, что в начале времён в храме города Бенарес бог Брахма создал три стержня и надел на один из них 64 диска. Монахи непрерывно переносят диски со стержня на стержень по вышеописанным правилам. Когда они перенесут все диски на другой стержень, наступит конец света. Используя результаты пунктов (а) и (б), определите, следует ли ожидать конца света в ближайшем будущем.

3. Элементы комбинаторики

Пусть $|X| = n,\ |Y| = m$.
Число всех функций $f : X \to Y$ (= число всех размещений m предметов по n ящикам) равно m^n.
Число всех слов длины n в m-буквенном алфавите равно m^n.
Число всех подмножеств множества X равно $|2^X| = 2^n$.
Число всех 1-1-функций $f : X \to Y$ (= число всех *размещений* m предметов по n ящикам, в которых каждый ящик содержит не более одного предмета) равно $A_m^n = m(m-1)\dots(m-n+1)$.
Число всех *перестановок* $f : X \to X$ равно $P(n) = n!$.
Число *сочетаний из n по k* (= число всех k-элементных подмножеств n-элементного множества) равно $C_n^k = A_n^k/k! = \frac{n!}{k!(n-k)!}$.
Бином Ньютона: $(x+y)^n = \sum\limits_{k=0}^{n} C_n^k x^k y^{n-k}$.

Задачи

3.1. Пусть $X = \{\, a, b, c \,\}$ – множество из трёх элементов.
а) Определите число двуместных функций, которые можно определить на X.
б) Определите число трёхместных отношений, которые можно определить на X.
в) Определите число трёхместных функций $f : X^3 \to X$, которые можно определить на X.

3.2. Предположим, что имеется n попарно непересекающихся множеств $A_1, A_2, \dots, A_n$ и $|A_i| = k_i, i = 1, \dots, n$. Докажите, что число множеств, содержащих не более чем по одному элементу из каждого A_i, равно $(k_1 + 1)(k_2 + 1) \cdots (k_n + 1)$.

3.3. Пусть $n = p_1^{k_1} p_2^{k_2} \cdots p_r^{k_r}$ — разложение n на простые сомножители. Используя предыдущую задачу, докажите, что число различных делителей n равно $t(n) = \prod_{i=1}^{r}(k_i + 1)$. Можно ли отсюда заключить, что n является квадратом тогда и только тогда, когда $t(n)$ нечётно?

3.4. В парламенте страны X имеется 401 место. Сколько имеется вариантов распределения мест в парламенте между тремя партиями, при которых ни одна из них не имеет абсолютного большинства?

3.5. Пусть $A = \{a, b, c, d, e, f, g, h\}$. Сколько существует
а) трёхэлементных подмножеств множества A?
б) пятиэлементных подмножеств множества A, содержащих b?
в) пятиэлементных подмножеств множества A, не содержащих b?
г) пятиэлементных подмножеств множества A, содержащих c, но не содержащих f и h?
д) подмножеств множества A, содержащих хотя бы три элемента?
е) подмножеств множества A, содержащих не более шести элементов?

3.6. Сколько имеется различных путей на плоскости из точки $(0, 0)$ в точку (x, y), проходящих через точки с целочисленными координатами, в которых каждый отрезок единичной длины идёт либо слева направо, либо снизу вверх?

3.7. Докажите формулу бинома Ньютона, используя метод математической индукции.

3.8. а) В разложении $(x + 2y)^7$ найдите коэффициент при x^3y^4.
б) В разложении $(5x - y)^6$ найдите коэффициент при x^3y^3.

3.9. Доказать тождества:
а) $nC_{n-1}^{k-1} = kC_n^k$;
б) $C_n^k C_{n-k}^{m-k} = C_m^k C_n^m$.

3.10. Доказать, что
$C_n^0 < C_n^1 < \ldots < C_n^{\lfloor n/2 \rfloor} = C_n^{\lceil n/2 \rceil} > C_n^{\lceil n/2 \rceil + 1} > \ldots > C_n^n$.

3.11. Сколько различных слов можно составить из букв слова АБРАКАДАБРА ?

3.12. Докажите, что число упорядоченных разбиений n-элементного множества на k подмножеств, первое из которых содержит n_1 элементов, второе — n_2 элементов, ..., k-ое — n_k элементов, равно $\frac{n!}{n_1!n_2!\ldots n_k!}$.

3.13. Преподаватель рассчитывает читать один и тот же курс в течение 20 лет. Чтобы не наскучить студентам, он решил рассказывать им каждый год 3 анекдота и не повторять никакие два года одни и те же 3 анекдота. Каково минимальное число анекдотов, которые он должен приготовить?

3.14. На острове N живет племя туземцев, у которых набор зубов во рту состоит из 30 зубов. При этом на острове нет двух жителей с одинаковыми наборами зубов. Может ли на острове N быть больше жителей чем в
а) Торжке? б) Твери? в) Москве? г) России? д) всём мире?

3.15. Сколько существует перестановок букв a, c, f, m, p, r, t, x, если
а) нет никаких ограничений?
б) между a и c должны стоять две или три буквы?
в) буквы a и c не должны быть разделены двумя или тремя буквами?
г) первые четыре буквы должны быть выбраны из a, c, f и r?
д) буквы a, c, f и r должны стоять рядом?

3.16. Две команды играют в волейбол до 3 побед. Сколько существует различных вариантов изменения счёта по партиям?

3.17. В кондитерском магазине продаются 4 сорта пирожных: заварные, песочные, “картошка” и бисквитные.
а) Сколькими способами можно купить 7 пирожных?
б) Сколькими способами можно купить 7 пирожных, если должно быть куплено хотя бы одно пирожное каждого сорта?

3.18. Сколько неотрицательных целочисленных решений имеет уравнение:

$$x + y + z + u = 10?$$

3.19. Сколько имеется слов длины n в алфавите $\{0, 1\}$, содержащих k единиц, в которых никакие две единицы не стоят рядом?

3.20. На книжной полке стоят n книг. Сколькими способами можно взять k книг, так чтобы не брать соседние книги и книги, стоящие на краях полки?

3.21. а) Сколько существует строго возрастающих последовательностей целых чисел, которые начинаются 1 и заканчиваются 7?
б) Сколько существует строго возрастающих последовательностей целых чисел, которые начинаются 4 и заканчиваются 11?
в) Сколько существует строго возрастающих последовательностей целых чисел, которые начинаются k и заканчиваются $k+m$?

3.22. Известно, что четверть пятиэлементных подмножеств множества $\{1, 2, 3, \ldots, n\}$ содержат элемент 7. Определите, чему равно n ($n \geq 5$).

3.23. Доказать *тождество Коши:*

$$C_{n+m}^{k} = \sum_{i=0}^{i=k} C_n^i C_m^{k-i}.$$

Указание: покажите, что обе части этого равенства задают количество вариантов выбора k человек из группы, состоящей из n женщин и m мужчин.

3.24. Докажите, что число способов, которыми можно породить k-элементное множество с повторениями, имея n разных элементов, например, $1, 2, \ldots, n$, из которых каждый может использоваться произвольное число раз, равно C_{n+k-1}^{k}. (Например, при $n = 5, k = 4$ множество $\{1, 2, 1, 3\}$ равно множеству $\{3, 1, 1, 2\}$ и не равно множеству $\{1, 2, 2, 3\}$).

3.25. В олимпиаде по программированию участвует n студентов. Награждаются 1-й, 2-й и 3-й премиями три победителя и еще 5 человек получают одинаковые дипломы. Сколько существует различных вариантов награждения?

3.26. Назовем два исхода первенства России по футболу *совпадающими в главном,* если в этих исходах совпадают обладатели золотых, серебряных и бронзовых медалей, а также две команды, покидающие премьер-лигу (т.е. занявшие два последних места). Найдите число *различных в главном* исходов (напомним, что в первенстве участвуют 16 команд).

3.27. N девушек водят хоровод. Сколькими различными способами они могут встать в круг?

3.28. Карточные игры.
а) При игре в бридж колоду из 52 карт раздают 4 игрокам — каждому

по 13 карт. Укажите, какие из следующих формул определяют, каким числом способов можно произвести такую раздачу?
Ответ 1. $4 * C_{52}^{13}$
Ответ 2. $52!/(4 * 13!)$
Ответ 3. $A_{52}^{13} * A_{39}^{13} * A_{26}^{13}$
Ответ 4. $C_{52}^{13} * C_{39}^{13} * C_{26}^{13}$
Ответ 5. $52!/C_{52}^{3}$.

б) При игре в преферанс колоду из 32 карт раздают трем игрокам — каждому по 10 карт, а оставшиеся 2 карты оставляют в прикупе. Укажите, какие из следующих формул определяют, каким числом способов можно произвести такую раздачу?
Ответ 1. $3 * C_{32}^{2} * C_{30}^{10}$
Ответ 2. $32! * C_{32}^{2}/(3 * 10!)$
Ответ 3. $C_{32}^{2} * C_{30}^{1}0 * C_{20}^{10}$
Ответ 4. $32!/(2! * 10! * 3)$
Ответ 5. $A_{32}^{30} * C_{32}^{2}$.

в) При игре в "дурака" колоду из 36 карт раздают четырем игрокам — каждому по 6 карт, а оставшиеся 12 карт оставляют в прикупе в фиксированном порядке. Далее в процессе игры карты из прикупа замещают в указанном порядке карты, выбывшие из игры, поэтому их порядок существенен. Укажите, какие из следующих формул определяют, каким числом способов можно произвести такую раздачу?
Ответ 1. $A_{36}^{12} * 24!/(4 * 6!)$
Ответ 2. $C_{36}^{12} * 24!/(4 * 6!)$
Ответ 3. $C_{36}^{12} * C_{30}^{10} * C_{20}^{10}$
Ответ 4. $A_{36}^{12} * 24!$
Ответ 5. $A_{36}^{12} * C_{24}^{6} * C_{18}^{6} * C_{12}^{6}$.

г) **Общий случай.** Пусть в некоторой карточной игре колоду из N карт раздают k игрокам — каждому по r карт, а оставшиеся $m = N - kr$ карт оставляют в прикупе. Выведите формулы, определяющие, каким числом способов можно произвести такую раздачу в случаях, когда
г.1) порядок карт в прикупе несущественен (как в преферансе) и
г.2) порядок карт в прикупе существенен (как в при игре в "дурака").

3.29. Пять юношей и пять девушек купили 10 билетов в кино в одном ряду. Сколькими способами они могут занять места, если

а) все юноши сядут рядом;
б) два юноши сядут по краям;
в) никакие два юноши не будут сидеть рядом;
г) ни юноши, ни девушки не будут сидеть все вместе;
д) один юноша и одна девушка всегда будут сидеть рядом;
е) один юноша и одна девушка откажутся сесть рядом.

3.30. За круглым столом короля Артура сидят 12 рыцарей. Каждый из них враждует со своими соседями. Нужно выбрать 5 рыцарей, чтобы освободить принцессу. Сколькими способами это можно сделать так, чтобы среди выбранных рыцарей не оказалось врагов?

Решите эту задачу в случае, когда из n рыцарей за столом нужно выбрать k рыцарей.

3.31. Установите принцип включения и исключения в теоретико-множественной форме.

Пусть $A_1, \ldots, A_n$ — это подмножества некоторого конечного множества X. Тогда

$$|\bigcup_{i=1}^{n} A_i| = \sum_{i=1}^{n} |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| -$$

$$\ldots + (-1)^{n-1} |A_1 \cap A_2 \cap \ldots \cap A_n|.$$

3.32. Сколько чисел в первой сотне не делится ни на одно из чисел 2, 3, 5? А в первой тысяче чисел?

3.33. Определите, сколько целочисленных решений имеет система:

$$\begin{cases} x + y + z = 11 \\ 0 \leq x \leq 4 \\ 0 \leq y \leq 3 \\ 0 \leq z \leq 7 \end{cases}$$

3.34. Докажите, что число слов длины n в алфавите из k букв, в которых каждая буква встречается хоть один раз равно

$$U^*(n, k) = \sum_{i=0}^{k-1} (-1)^i C_k^i (k - i)^n.$$

3.35. У заведующего лабораторией имеется 5 сотрудников. Сколькими способами он может распределить среди них 8 различных заданий так, чтобы у каждого было хоть одно задание?

3.36. Найти число перестановок из n элементов, при которых ни один элемент не остается в первоначальном положении.

4. Булевы функции и их представления

$B = \{0, 1\}$. *Единичный n-мерный куб*: $B^n = \underbrace{B \times B \times \ldots \times B}_{n}$ — это множество всех *двоичных последовательностей (векторов)* длины n.
n-местная булева функция: $f(x_1, \ldots, x_n) : B^n \to B$.
$\mathcal{P}_n$ — это множество всех n-местных булевых функций.
При $n = 0$ $\mathcal{P}_0$ состоит из двух 0-местных функций-констант 0 и 1.

Пусть $\mathcal{B}$ — некоторое (конечное или бесконечное) множество булевых функций. $V = \{X_1, X_2, \ldots\}$ — счётное множество переменных. Определим по индукции множество *формул* над $\mathcal{B}$ с переменными из V. Одновременно будем определять числовую характеристику $dep(\Phi)$ формулы Φ, называемую её *глубиной*.
а) *Базис индукции.* Каждая переменная $X_i \in V$ и каждая константа $c \in \mathcal{B}$ является формулой глубины 0, т.е. $dep(X_i) = dep(c) = 0$.
б) *Шаг индукции.* Пусть $f(x_1, \ldots, x_m) \in \mathcal{B}$, $A_1, \ldots, A_m$ — формулы, и $\max\{dep(A_i) \mid i = 1, \ldots, m\} = k$. Тогда выражение $\Phi = f(A_1, \ldots, A_m)$ является формулой, её глубина $dep(\Phi)$ равна $k + 1$.
Для формул над множеством функций $\mathcal{B}_e = \{0, 1, \neg, \wedge, \vee, \rightarrow, +, \sim, \ |, \downarrow\}$, представленных в таблицах 1 и 2, будем использовать *инфиксную запись*, в которой имя 2-местной функции (логической связки) помещается между 1-м и 2-м аргументами.

Формула называется *выполнимой*, если при некоторых значениях входящих в неё переменных она равна 1.
Формула называется *тождественно истинной*, если при любых значениях входящих в неё переменных она равна 1.
Длина формулы — это число входящих в неё символов функций и переменных.

Таблица 1. Булевы функции от одной переменной

x_1	0	1	x_1	$\neg x_1$
0	0	1	0	1
1	0	1	1	0

Таблица 2. Основные булевы функции от двух переменных

x_1	x_2	$\wedge$	$\vee$	$\rightarrow$	$+$	$\sim$	$\mid$	$\downarrow$
0	0	0	0	1	0	1	1	1
0	1	0	1	1	1	0	1	0
1	0	0	1	0	1	0	1	0
1	1	1	1	1	0	1	0	0

Задачи

4.1. а) Сколько имеется k-мерных граней в n-мерном кубе?
б) Каково общее число граней n-мерного куба?

4.2. Пусть $\tilde{a} = (a_1, a_2, \ldots, a_n)$ и $\tilde{b} = (b_1, b_2, \ldots, b_n)$ – два набора (две точки) n-мерного единичного куба B^n. Расстоянием между $\tilde{a}$ и $\tilde{b}$ *по Хэммингу* называется число $d(\tilde{a}, \tilde{b}) = \sum\limits_{i=1}^{n} |a_i - b_i|$. Докажите, что это расстояние удовлетворяет следующим свойствам для всех $\tilde{a}, \tilde{b}$ и $\tilde{c}$ из B^n:
а) неотрицательность: $d(\tilde{a}, \tilde{b}) \geq 0$ и $(d(\tilde{a}, \tilde{b}) = 0 \iff \tilde{a} = \tilde{b})$;
б) симметричность: $d(\tilde{a}, \tilde{b}) = d(\tilde{b}, \tilde{a})$;
в) неравенство треугольника: $d(\tilde{a}, \tilde{b}) \leq d(\tilde{a}, \tilde{c}) + d(\tilde{c}, \tilde{b})$.

4.3. Пусть $\tilde{a}$ и $\tilde{b}$ — две точки n-мерного единичного куба B^n и $d(\tilde{a}, \tilde{b}) = k$. Докажите, что тогда
а) самый короткий путь по рёбрам B^n, соединяющий $\tilde{a}$ и $\tilde{b}$ содержит k рёбер;
б) всего существует $k!$ различных таких путей.

4.4. Для каждой из следующих функций определите соответствующее ей множество вершин 4-мерного единичного куба B^4 $N_f^+ = \{(a_1, a_2, a_3, a_4) \mid f(a_1, a_2, a_3, a_4) = 1\}$ и найдите его мощность: а) $f_1(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow x_1 = 0$;
б) $f_2(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow x_4 = 1$;
в) $f_3(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow x_1 + x_2 \geq x_3 + x_4$ (здесь $+$ — арифметическое

сложение);
г) $f_4(x_1, x_2, x_3, x_4) = 1 \Leftrightarrow\ x_1x_2 = 0$ или $x_3x_4 = 1$.

4.5. Построить таблицы значений для следующих булевых функций:
а) $f_1(X_1, X_2, X_3) = 1\ \Leftrightarrow\ X_1 + X_3 \geq X_2$;
б) $f_2(X_1, X_2, X_3) = 1\ \Leftrightarrow$ сумма $(X_1 + X_2 + X_3)$ чётна;
в) $f_3(X_1, X_2, X_3) = 0\ \Leftrightarrow$ значение X_1 совпадает со значением X_2 или со значением X_3.
г) $f_4(X_1, X_2, X_3) =$ если $X_1 = 1$, то X_2, иначе X_3.

4.6. Для каждой из следующих формул определить её глубину и построить таблицу задаваемой ею функции.
а) $\Psi_1 = ((X_1 \to \neg X_3) \vee (X_2 + X_3))$;
б) $\Psi_2 = (\neg(X_1 \mid X_2) \sim (\neg X_1 \wedge X_2))$;
в) $\Psi_3 = ((X_2 + \neg X_3) \wedge ((X_1 \vee X_2) \to (X_1 \sim \neg X_3)))$.
г) $\Psi_4 = ((\neg X_1 \wedge X_2) \downarrow (X_1 \to \neg X_3))$.
д) $\Psi_5 = (((X_1 + X_2) + X_3) \mid (\neg X_2 \downarrow X_3))$.

4.7. В представленной ниже таблице показано 4-битовое кодирование десятичных цифр от 0 до 9.

Цифра	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
...	.	.	.	.
9	1	0	0	1
Ошибочные	1	0	1	0
	.	.	.	.
оды	1	1	1	1

Какие из следующих булевых формул задают множество всех ошибочных кодов?
а) $(\neg A \to (B \wedge C))$
б) $((A \wedge B) \vee (C \wedge D))$
в) $((A \wedge B) \vee (A + C))$
г) $((A \wedge \neg B) \vee (C \wedge D))$
д) $((A \wedge B) \vee (A \wedge C))$

4.8. Назовём два набора $\alpha = (\alpha_1, \ldots, \alpha_n) \in B^n$ и $\beta = (\beta_1, \ldots, \beta_n) \in B^n$ *соседними*, если они находятся в соседних строках таблицы для функции от n переменных, т.е. представляют двоичные записи чисел i_α и i_β, для которых $|i_\alpha - i_\beta| = 1$.

Найти число функций в $\mathcal{P}_n$, которые на любой паре соседних наборов принимают
а) одинаковые значения;
б) разные значения.

4.9. Назовём два набора $\alpha = (\alpha_1, \ldots, \alpha_n) \in B^n$ и $\beta = (\beta_1, \ldots, \beta_n) \in B^n$ *противоположными*, если для всякого i $\alpha_i = 1$ тогда и только тогда, когда $\beta_i = 0$ (и, следовательно, $\alpha_i = 0$ тогда и только тогда, когда $\beta_i = 1$).

Найти число функций в $\mathcal{P}_n$, которые на любой паре противоположных наборов принимают разные значения.

4.10. Пусть $n = 2k$. Назовем набор $\alpha = (\alpha_1, \ldots, \alpha_n) \in B^n$ ***парным***, если для любого $i = 1, \ldots, k$ $\alpha_i = \alpha_{k+i}$, т.е. $\alpha = \alpha'\alpha'$ для некоторого набора α' размера k.

Найти число функций в $\mathcal{P}_n$, которые на всех парных наборах принимают одинаковое значение.

4.11. Какие из следующих условий можно выразить булевыми формулами от переменных p_1, p_2, p_3, p_4, использующими лишь логические связки $\wedge$ и $\vee$ (без отрицания $\neg$)?
а) По крайней мере две переменные из p_1, p_2, p_3, p_4 истинны (равны 1).
б) Не все из переменных из p_1, p_2, p_3, p_4 ложны (равны 0).
в) Нечётное число переменных из p_1, p_2, p_3, p_4 истинны (равны 1).

4.12. Детектив Ш. Холмс подозревает в совершении преступления трех лиц: A, B и C. Он установил, что
а) если B преступник, то и C является преступником;
б) кто-то один из пары A, C является преступником (но не оба вместе);
в) если C не преступник, то и A не преступник.
Опишите знания Ш. Холмса в виде булевой формулы и постройте таблицу её значений. Может ли он сделать вывод, что C является преступником? Можно ли достоверно утверждать, что преступник действовал в одиночку?

4.13. Детектив Э. Пуаро подозревает в совершении преступления трех лиц: A, B и C. Они дали следующие показания:
A : если B преступник, то C не виновен;
B : если A виновен, то и C является преступником;
C : A преступник.
Э. Пуаро установил, что если A сказал правду, то B соврал, и что показаниям C нельзя доверять. Опишите знания Э. Пуаро в виде булевой

формулы и постройте таблицу её значений. Может ли он сделать вывод, что B является преступником? Мог ли преступник быть один?

4.14. Администратор базы данных обнаружил, что одна или несколько из трёх записей его базы A, B и C ошибочна. Он установил, что
а) если запись B корректна, то A ошибочна;
б) хотя бы одна запись из пары B, C корректна и хотя бы одна запись из пары A, C корректна;
в) если A ошибочна, то хотя одна из записей B, C корректна (но не обе вместе).
Опишите знания администратора в виде булевой формулы. Может ли он сделать вывод, что запись B ошибочна? Можно ли достоверно утверждать, что ошибочная запись единственна?

4.15. Программист Пётр использовал в своей программе три целочисленные переменные x, y и z. В определённом месте программы он поместил условный оператор:
IF $(x * y \geq 0)$ OR $(x * z \geq 0)$ THEN $x = 1$ ELSE $x = 2$;

Проанализировав свою программу, Пётр установил, что перед выполнением этого оператора выполнены следующие условия:
а) если $z < 0$, то $x < 0$ или $y \geq 0$;
б) $x \geq 0$ или $y < 0$;
в) если $y < 0$, то хотя бы одна из переменных x, z отрицательна, но не обе вместе.

Опишите знания Петра в виде булевой формулы. Может ли он оптимизировать программу, заменив указанный условный оператор на присваивание $x = 1$ или на присваивание $x = 2$? Если "да", то на какое?

4.16. Комитет состоит из пяти членов. Решения принимаются большинством голосов, однако, если председатель голосует "против", то решение не принимается. Постройте формулу, зависящую от 5 переменных X_1, X_2, X_3, X_4, Y ($X_i = 1$ тогда и только тогда, когда i-ый член комитета голосует "за", $Y = 1$ тогда и только тогда, когда председатель "за"), значение которой равно 1 тогда и только тогда, когда в результате голосования решение принимается.

4.17. Какие из следующих формул являются тождественно истинными?
$A = ((X \to \neg Y) \to ((\neg Y \to Z) \to (X \to Z)))$,
$B = ((X \to \neg Y) \to ((X \to Z) \to (\neg Y \to Z)))$,
$C = ((X \to \neg Y) \to ((X \to (\neg Y \to Z)) \to (X \to Z)))$,
$D = ((\neg X \to Y) \to ((\neg Y \to Z) \to (X \to Z)))$.

4.18. Пусть во всех зоопарках, где есть львы и носороги, нет жирафов; во всех зоопарках, где есть носороги и нет жирафов, есть львы; наконец, во всех зоопарках, где есть львы и жирафы, есть и носороги. Как вы думаете, может ли существовать такой зоопарк, в котором есть львы, но нет ни жирафов, ни носорогов?

4.19. Анна, Беатриса и Вера как-то обнаружили, что все они в одинаковых джинсах. Известно, что у Анны есть джинсы с карманами, узкие джинсы и вылинявшие джинсы без карманов; у Беатрисы — джинсы без карманов и вылинявшие узкие джинсы с карманами. И наконец, Вера имеет джинсы-клёши и синие узкие джинсы с карманами. Как выглядят их одинаковые джинсы?

4.20. К сожалению, не все наши студенты в совершенстве владеют английским языком. Один из них, попавший на стажировку в Англию и несколько поиздержавшийся там, должен был попасть в Манчестер. Он проголосовал на шоссе и остановил машину, где находились отец, мать и дочь, которые ответили ему на вопрос об их поездке. Каждая произнесенная кем-то из них фраза имела два смысла, и он не мог решить, каков смысл фразы на самом деле. Вот, что они говорили (второй возможный смысл указан в скобках).
Отец: мы отправляемся в Манчестер (мы едем из Ньюкасла).
Мать: мы не отправляемся в Манчестер, а едем из Ньюкасла (мы не остановились в Ливерпуле и едем из Ньюкасла).
Дочь: мы не едем из Ньюкасла (мы остановились в Ливерпуле).
Следует ли студенту садиться в эту машину?

4.21. *Задача Льюиса Кэрролла.*
Автор широко известной книги “Алиса в Стране чудес” любил задавать следующую задачу из четырёх фраз: “Из двух одно: или злоумышленник уехал в экипаже, или свидетель ошибся. Если злоумышленник имел сообщника, то он уехал в экипаже. У злоумышленника не было ни сообщника, ни ключа или у него был сообщник и был ключ. У злоумышленника был ключ”. Какой вывод отсюда можно сделать?

4.22. Профессор экономики сформулировал студентам три правила:
1) если инвестиции не увеличатся, то возрастут правительственные расходы или возникнет безработица;
2) если правительственные расходы не возрастут, то налоги снизятся;
3) если налоги снизятся и инвестиции увеличатся, то не возникнет безра-

ботица.
Оказалось, что в данный момент инвестиции не увеличились.

Могут ли студенты сделать вывод о том, что *правительственные расходы возрастут*?

5. Эквивалентность формул

Основные эквивалентности (законы логики):
1) Ассоциативность.
$((X_1 \circ X_2) \circ X_3) \equiv (X_1 \circ (X_2 \circ X_3))$, где $\circ \in \{\wedge, \vee, +\}$
2) Коммутативность.
$(X_1 \circ X_2) \equiv (X_2 \circ X_1)$, где $\circ \in \{\wedge, \vee, +\}$
3) Дистрибутивные законы.
$((X_1 \vee X_2) \wedge X_3) \equiv ((X_1 \wedge X_3) \vee (X_2 \wedge X_3))$
$((X_1 \wedge X_2) \vee X_3) \equiv ((X_1 \vee X_3) \wedge (X_2 \vee X_3))$
$((X_1 \wedge X_2) + X_3) \equiv ((X_1 \wedge X_3) + (X_2 \wedge X_3))$
4) Двойное отрицание.
$\neg\neg X \equiv X$
5) Законы де Моргана.
$\neg(X_1 \vee X_2) \equiv (\neg X_1 \wedge \neg X_2)$
$\neg(X_1 \wedge X_2) \equiv (\neg X_1 \vee \neg X_2)$
6) Законы упрощения.

$(X \wedge X) \equiv X$	$(X \vee X) \equiv X$
$(X \wedge \neg X) \equiv 0$	$(X \vee \neg X) \equiv 1$
$(X \wedge 0) \equiv 0$	$(X \vee 0) \equiv X$
$(X \wedge 1) \equiv X$	$(X \vee 1) \equiv 1$

7) $(X_1 \rightarrow X_2) \equiv (\neg X_1 \vee X_2)$
8) $(X_1 + X_2) \equiv ((X_1 \wedge \neg X_2) \vee (\neg X_1 \wedge X_2))$
Законы поглощения.
П1) $X \vee (X \wedge \Phi) \equiv X$
П2) $(X \wedge \Phi) \vee (\neg X \wedge \Phi) \equiv \Phi$
П3) $(X_1 \wedge X_2) \vee (\neg X_1 \wedge X_3) \vee (X_2 \wedge X_3) \equiv (X_1 \wedge X_2) \vee (\neg X_1 \wedge X_3)$

Задачи

5.1. Проверьте все приведенные выше эквивалентности (1)-(8), непосредственно вычисляя функции, представляемые их левыми и правыми частями.

5.2. Выведите законы поглощения П1, П2 и П3, используя основные эквивалентности.

5.3. Назовём *логическим произведением* формулу вида $\Phi_1 \wedge \Phi_2 \wedge \ldots \wedge \Phi_n$ (в этом выражении использованы соглашения о сокращении записи!). Её подформулы $\Phi_i, 1 \leq i \leq n$, будем называть *сомножителями*. Аналогично, *логической суммой* назовём формулу вида $\Phi_1 \vee \Phi_2 \vee \ldots \vee \Phi_n$. Её подформулы $\Phi_i, 1 \leq i \leq n,$, будем называть *слагаемыми*.

Покажите, что из основных тождеств можно вывести следующие правила преобразования логических произведений и сумм.
С1) Если в логическом произведении один из сомножителей равен 0, то и все произведение равно 0.
С2) Если в логической сумме одно из слагаемых равно 1, то и вся сумма равна 1.
С3) Если в логическом произведении $n \geq 2$ и есть сомножитель, равный 1, то его можно вычеркнуть.
С4) Если в логической сумме $n \geq 2$ и есть слагаемое, равное 0, то его можно вычеркнуть.

5.4. Используя основные тождества, доказать эквивалентность следующих пар формул.
а) $\neg(X \vee \neg Y) \wedge (X \to \neg Y)$ и $(\neg X \wedge Y)$;
б) $\neg[(X \wedge \neg Y) \to (\neg X \vee Z)]$ и $(X \wedge \neg Y \wedge \neg Z)$;
в) $(X + Y) \to (X \wedge \neg Y)$ и $(\neg X \wedge \neg Y) \vee X$.

5.5. Используя основные тождества, доказать тождественную истинность следующих формул:
а) $(A \to (B \to A))$;
б) $((A \wedge B) \to A)$;
в) $(((A \to B) \wedge (B \to C)) \to (A \to C))$;
г) $(A \to (A \vee B))$;
д) $(((A \to B) \vee (A \to C)) \to ((A \to (B \vee C)))$;
е) $((A \to B) \to ((A \to \neg B) \to \neg A))$;
ж) $((A \to \neg B) \to \neg(A \wedge B))$.

5.6. Булева функция $f^*(X_1, \ldots, X_n)$ называется *двойственной* к функции $f(X_1, \ldots, X_n)$, если $f^*(X_1, \ldots, X_n) = \neg f(\neg X_1, \ldots, \neg X_n)$. Например, $X \wedge Y$ двойственна к $X \vee Y$ и наоборот.

а) Докажите, что отношение *двойственности* симметрично.

б) Пусть $f(Y_1, \ldots, Y_m), g_1(X_1, \ldots, X_n), \ldots, g_m(X_1, \ldots, X_n)$ — булевы функции и
$F(X_1, \ldots, X_n) = f(g_1(X_1, \ldots, X_n), \ldots, g_m(X_1, \ldots, X_n))$.

Установите следующий *принцип двойственности:* двойственная функция от суперпозиции функций равна суперпозиции двойственных функций:
$F^*(X_1, \ldots, X_n) = f^*(g_1^*(X_1, \ldots, X_n), \ldots, g_m^*(X_1, \ldots, X_n))$.

5.7. Построить двойственные функции для функций $\neg, \to, +, |, \downarrow$.

5.8. Построить двойственные функции для функций, заданных следующими формулами:
а) $(X \to \neg(\neg Y + X + Z))$;
б) $((X|Y) + (\neg X \to Z))$;
в) $((X \vee \neg Y) \to (\neg X \downarrow Y))$.

5.9. а) Докажите, что среди 20 формул от переменных X_1, X_2 всегда найдутся две эквивалентные.
б) Верно ли, что среди 20 формул от переменных X_1, X_2 всегда найдутся три эквивалентные?
в) Пусть имеется k формул от n переменных $X_1, \ldots, X_n$. Найдите наименьшее число эквивалентных формул в этом множестве.

6. Дизъюнктивные и конъюнктивные нормальные формы

Пусть $\mathbf{X} = \{X_1, \ldots, X_n\}$ — это множество переменных.
Пусть $X_i^0 = \neg X_i$ и $X_i^1 = X_i$.
Элементарная конъюнкция (элементарная дизъюнкция) — это формула вида $X_{i_1}^{\sigma_1} \wedge X_{i_2}^{\sigma_2} \wedge \ldots \wedge X_{i_k}^{\sigma_k}$ $(X_{i_1}^{\sigma_1} \vee X_{i_2}^{\sigma_2} \vee \ldots \vee X_{i_k}^{\sigma_k})$, где $k \geq 1$.
Дизъюнктивная нормальная форма (ДНФ) $\mathcal{D}$ — это дизъюнкция элементарных конъюнкций: $\mathcal{D} = K_1 \vee K_2 \vee \ldots \vee K_r$, где K_j $(j = 1, ..., r)$ — это элементарная конъюнкция.
$\mathcal{D}$ называется *совершенной ДНФ*, если в каждую K_j входят все n переменных из $\mathbf{X}$.
Конъюнктивная нормальная форма (КНФ) $\mathcal{C}$ — это конъюнкция элементарных дизъюнкций:
$\mathcal{C} = D_1 \wedge D_2 \wedge \ldots \wedge D_r$, где D_j $(j = 1, ..., r)$ — это элементарная дизъюнкция.
$\mathcal{C}$ называется *совершенной КНФ*, если в каждую D_j входят все n переменных из $\mathbf{X}$.

Пусть $f(x_1, \ldots, x_n)$ — булева функция.
$N_f^+ = \{(\sigma_1, \sigma_2, \ldots, \sigma_n) \mid f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1\}$,
$N_f^- = \{(\sigma_1, \sigma_2, \ldots, \sigma_n) \mid f(\sigma_1, \sigma_2, \ldots, \sigma_n) = 0\}$.
Для элементарной конъюнкции K N_K^+ — это множество наборов значений переменных, на которых K принимает значение 1.

Элементарная конъюнкция K называется *допустимой для* f, если $N_K^+ \subseteq N_f^+$.
Допустимая элементарная конъюнкция K называется *максимальной для* f, если для любой элементарной конъюнкции L из условия $N_K^+ \subseteq N_L^+ \subseteq N_f^+$ следует, что $N_K^+ = N_L^+$.
Сокращённой ДНФ для функции f называется дизъюнкция всех максимальных для этой функции элементарных конъюнкций.
Метод Блейка для построения сокращённой ДНФ, эквивалентной ДНФ D:
(1) Применять к D, сколько возможно, слева направо закон поглощения (П3): $(X \wedge K_1) \vee (\neg X \wedge K_2) \equiv (X \wedge K_1) \vee (\neg X \wedge K_2) \vee (K_1 \wedge K_2)$.
(2) Применять, сколько возможно, закон поглощения (П1): $X \vee (X \wedge K) \equiv X$.
(3) Упростить полученную формулу.
Минимальная ДНФ для функции f — это её ДНФ, имеющая минимальную длину.

Задачи

6.1. В теореме о построении совершенных ДНФ и КНФ для функции $f(X_1, \ldots, X_n)$ были построены две формулы: $\mathcal{D}_f = \bigvee_{(\sigma_1,\ldots,\sigma_n) \in N_f^+} (X_1^{\sigma_1} \wedge X_2^{\sigma_2} \wedge \cdots \wedge X_n^{\sigma_n})$ и
$\mathcal{C}_f = \bigwedge_{(\sigma_1,\ldots,\sigma_n) \in N_f^-} (X_1^{\neg\sigma_1} \vee X_2^{\neg\sigma_2} \vee \cdots \vee X_n^{\neg\sigma_n})$. Докажите, что формулы $\mathcal{D}_f$ и $\mathcal{C}_f$ являются совершенной ДНФ и КНФ для функции f, проверив, что для любого набора значений аргументов $\sigma_1, \ldots, \sigma_n$ выполнены равенства $f(\sigma_1, \ldots, \sigma_n) = \mathcal{D}_f(\sigma_1, \ldots, \sigma_n)$ и $f(\sigma_1, \ldots, \sigma_n) = \mathcal{C}_f(\sigma_1, \ldots, \sigma_n)$.

6.2. Наборы значений трёх аргументов X, Y и Z булевых функций f и g упорядочены лексикографически. Их значения задаются следующими последовательностями 8 нулей и единиц: $f = (1011\ 0011)$, $g = (0011\ 1001)$. Определите для каждой из функций f и g совершенную дизъюнктивную и совершенную конъюнктивную нормальные формы.

6.3.
а) Предложите процедуру, которая по произвольной элементарной конъюнкции строит эквивалентную ей совершенную ДНФ.
б) Предложите процедуру, которая по произвольной элементарной дизъюнкции строит эквивалентную ей совершенную КНФ.

6.4. Используя основные эквивалентности, постройте для следующих формул эквивалентные совершенные ДНФ и КНФ:
а) $\Phi_1 = ((X \mid Y) \vee (Z + X)) \to \neg Y$;
б) $\Phi_2 = ((\neg X + Z) \to (X \vee Y)) \wedge (\neg X \to \neg Y)$;
в) $\Phi_3 = (((Z \vee X) + Y) \vee (X \to \neg Z)) \wedge Z$.

6.5. Преобразовать следующие ДНФ в эквивалентные совершенные ДНФ:
а) $(X \wedge Y) \vee (\neg X \wedge Z) \vee (X \wedge Z)$;
б) $(\neg X \wedge Y) \vee Z \vee (X \wedge \neg Z)$;
в) $X \vee (\neg X \wedge Z) \vee (Y \wedge Z) \vee (\neg X \wedge \neg Y)$.

6.6. Преобразовать следующие КНФ в эквивалентные совершенные КНФ:
а) $(X \vee Y) \wedge (\neg X \vee Z) \wedge (X \vee \neg Z)$;
б) $\neg X \wedge (X \vee Y) \wedge Z \wedge (X \vee \neg Z)$;
в) $(\neg X \vee Z) \wedge (Y \vee Z) \wedge (\neg X \vee \neg Y)$.

6.7. С помощью основных эквивалентностей преобразовать следующие ДНФ в КНФ:
а) $(X \wedge Y) \vee (\neg X \wedge Y \wedge Z) \vee (X \wedge Z)$;
б) $(\neg X \wedge \neg Y \wedge Z) \vee (\neg X \wedge Y \wedge Z) \vee (X \wedge Y \wedge \neg Z)$;
в) $X \vee (\neg X \wedge Z) \vee (\neg X \wedge Y \wedge Z) \vee (\neg X \wedge \neg Y \wedge \neg Z)$.

6.8. Докажите, что на этапе (1) процедуры "Приведение к совершенной ДНФ" при последовательном выполнении эквивалентных преобразований вида (7), а затем — вида (8) до тех пор, пока ни одно из них не применимо, полученная в результате формула не будет содержать функций $\to$ и $+$.

6.9. Как изменить (3)-й, (4)-й и (5)-й этапы процедуры "Приведение к совершенной ДНФ", чтобы в результате получить процедуру "Приведение к совершенной КНФ", которая по произвольной формуле строит эквивалентную совершенную КНФ?

6.10. Докажите, что для любого $k \leq n$ каждую булеву функцию $f \in \mathcal{P}_n$ можно представить в виде
$f(X_1, \ldots, X_k, X_{k+1}, \ldots, X_n) =$

$$\bigvee_{(\sigma_1,\ldots,\sigma_k)\in B^k} X_1^{\sigma_1} \wedge \ldots X_k^{\sigma_k} \wedge f(\sigma_1, \ldots, \sigma_k, X_{k+1}, \ldots, X_n).$$

Такое представление называется разложением f по $X_1, \ldots, X_k$.

6.11. а) Сколько неэквивалентных элементарных конъюнкций длины k можно составить из n переменных $X_1, \ldots, X_n$ и их отрицаний?
б) Сколько всего неэквивалентных элементарных конъюнкций можно составить из n переменных $X_1, \ldots, X_n$ и их отрицаний?

6.12. Определите, какие из следующих элементарных конъюнкций являются допустимыми и какие из них — максимальными для функции $f(X, Y, Z)$, заданной следующей последовательностью 8 нулей и единиц: $f = (10111010)$?
а) $\neg X \wedge Y \wedge Z$; б) $\neg Z$; в) $\neg X \wedge Y$;
г) $\neg Y$; д) $X \wedge \neg Z$.

6.13. Докажите, что совершенная, сокращённая и минимальная ДНФ для функции $odd(X_1, \ldots, X_n) = X_1 + \ldots + X_n$ совпадают и состоят из 2^{n-1} элементарных конъюнкций длины n.

6.14. Наборы значений трёх аргументов X, Y и Z булевых функций f и g упорядочены лексикографически. Их значения задаются следующими последовательностями 8 нулей и единиц: $f = (1110\ 1011)$, $g = (0011\ 1011)$. Определите для каждой из функций f и g сокращённую дизъюнктивную нормальную форму.

6.15. Используя основные эквивалентности, найти эквивалентные сокращённые ДНФ и доказать эквивалентность следующих пар формул:
а) $\Phi = (\neg(X \to (\neg Y \to (X \wedge \neg Z))) \wedge (Z \vee \neg(X \wedge Y)))$,
$\Psi = ((X \wedge Z) + (X \wedge Y \wedge Z))$;
б) $\Phi = (((X \wedge Y) \to \neg Z) \wedge (\neg X \to \neg Y))$,
$\Psi = (Y \to (X \wedge (Z + 1)))$;
в) $\Phi = (((X \vee Y) \to \neg Z) \wedge ((X \wedge Z) \to Y))$,
$\Psi = (Z \to ((X + 1) \wedge \neg Y))$.

6.16. Найдите эквивалентные сокращённые ДНФ для следующих формул. Какие из этих пар формул эквивалентны?
а) $\Phi = (((\neg X \wedge \neg Y) \to \neg Z) \wedge (X \to Y))$,
$\Psi = ((1 + Y) \to (\neg X \wedge (1 + Z)))$;
б) $\Phi = (\neg((X_1 \to X_2) \vee \neg(X_2 \to X_1)) \wedge X_3)$,
$\Psi = \neg((X_1 \wedge X_3) \to X_2)$;
в) $\Phi = \neg(\neg X \wedge Y \wedge \neg Z) \to ((Y + 1) \wedge ((X + 1) \to \neg(\neg U \vee \neg Z)))$,
$\Psi = (\neg X \vee Y) \to ((\neg U \vee Y \vee Z) \to (\neg(X \vee \neg Y) \wedge \neg Z))$.

7. Многочлены Жегалкина

Многочленом Жегалкина называется формула над множеством функций $F_J = \{0, 1, *, +\}$ (здесь $*$ — это другое обозначение конъюнкции), которая (возможно, после раскрытия скобок и "приведения" подобных членов) представляет сумму (по модулю 2) *положительных* элементарных конъюнкций.
Эквивалентности для преобразования ДНФ и КНФ в многочлены Жегалкина:

$$\begin{array}{lrcl}
(J1) & \neg X & \equiv & (X + 1), \\
(J2) & (X_1 \wedge X_2) & \equiv & (X_1 * X_2), \\
(J3) & (X_1 \vee X_2) & \equiv & (X_1 * X_2 + X_1 + X_2) \equiv \\
 & & \equiv & (X_1 + 1) * (X_2 + 1) + 1, \\
(J4) & (X_1 + X_2) * (X_3 + X_4) & \equiv & (X_1 * X_2 + X_1 * X_3 + X_2 * X_3 + X_2 * X_4).
\end{array}$$

Пусть $K_i = \bigwedge_{\sigma_i^j = 1} X_j$, где $i = \sigma_1 \sigma_2 \ldots \sigma_n$ — двоичное n-битовое представление i (в частности, $K_0 = 1$). Тогда стандартное представление многочлена Жегалкина для функции f от n переменных имеет вид
$f(X_1, \ldots, X_n) = \alpha_0 * K_0 + \alpha_1 * K_1 + \alpha_2 * K_2 + \ldots + \alpha_{2^n - 1} * K_{2^n - 1}$ $(\alpha_i \in \{0, 1\})$.
Метод неопределённых коэффициентов: подставляя в это равенство всевозможные значения переменных, получаем треугольную систему 2^n линейных уравнений относительно 2^n неизвестных коэффициентов α_i. Решая её, находим многочлен Жегалкина для f.

Задачи

7.1. Используя основные эквивалентности и тождества (J1)-(J4), найти эквивалентные многочлены Жегалкина и доказать эквивалентность следующих пар формул:
а) $\Phi = ((Z \wedge (X \to Y)) \vee \neg(\neg X \to Z))$, $\quad \Psi = (X \to (Y \wedge Z))$;
б) $\Phi = \neg(X \to (Y \wedge Z)) \wedge (\neg Y \vee \neg X)$, $\quad \Psi = (\neg(X \to Y) \vee Z) \wedge \neg((\neg X \wedge Z) \vee (Y \wedge Z))$;
в) $\Phi = (((Y \wedge Z) \to \neg(X \vee Z)) \wedge \neg(\neg Y \wedge Z \wedge X))$, $\quad \Psi = (Z \to (\neg Y \wedge \neg X))$;
г) $\Phi = \neg(((X_1 \to X_2) \vee \neg X_3) \wedge X_1)$, $\quad \Psi = (\neg(X_3 \to X_2) \vee \neg X_1)$;
д) $\Phi = (\neg((X_1 \to X_2) \vee \neg(X_2 \to X_1)) \wedge X_3)$, $\quad \Psi = \neg((X_1 \wedge X_3) \to X_2)$.

7.2. Найти многочлены Жегалкина (методом неопределённых коэффициентов) для следующих функций от трёх аргументов. Считаем, что наборы их аргументов упорядочены лексикографически и значения на них задаются последовательностью 8 нулей и единиц.
а) $f = (0010\ 1100)$, $\quad$ в) $f = (1100\ 0011)$,
б) $f = (1110\ 1100)$, $\quad$ г) $f = (0110\ 1011)$.

7.3. Найти методом неопределённых коэффициентов многочлены Жегалкина для следующих функций:
а) $f = (X \mid Y) \to (\neg Y \vee Z)$;
б) $g = (1010\ 1100\ 0110\ 1101)$;
в) $h = (X \wedge \neg Y \wedge \neg Z) \vee (\neg X \to (Y \wedge Z \wedge \neg U))$.

7.4. Докажите, что многочлен Жегалкина для функции $f(X_1, X_2, \ldots, X_n)$ можно получить, раскрыв скобки и упростив выражение

$$p_f(X_1, X_2, \ldots, X_n) = \sum_{(\sigma_1, \sigma_2, \ldots, \sigma_n) \in N_f^+} (X_1 + \sigma_1 + 1) * (X_1 + \sigma_2 + 1) * \ldots * (X_n + \sigma_n + 1).$$

7.5. Используя предыдущую задачу, найти многочлены Жегалкина для следующих функций:
а) $f = (0110\ 1000)$;
б) $g = (1010\ 1100)$;
в) $h = (X \wedge \neg Y \wedge Z) \to (Y \wedge Z)$.

7.6. Докажите, что многочлен Жегалкина тождественно равен нулю тогда и только тогда, когда все его коэффициенты в стандартном представлении равны нулю.

7.7. Длиной многочлена Жегалкина называется число его слагаемых (элементарных конъюнкций). Например, $p(X_1, X_2) = 1 + X_1 + X_1 * X_2$ имеет длину 3. Определите, сколько существует различных многочленов Жегалкина от n переменных длины k, которые обращаются в 0 на наборах $(0, 0, \ldots, 0)$ и $(1, 1, \ldots, 1)$, состоящих из одних нулей и единиц, соответственно.

7.8. Пусть длина многочлена Жегалкина определена как в предыдущей задаче. Доказать, что для всякого k $(k \leq 2^n)$ существует многочлен Жегалкина $P(X_1, \ldots, X_n)$ длины не большей чем n, для которого число единиц $|N_P^+| = k$. (Указание: используйте индукцию по n.)

7.9. Пусть многочлен Жегалкина задается формулой:
$P_n(X_1, \ldots, X_n) = 1 + X_1 + X_1 * X_2 + \ldots + X_1 * X_2 * \ldots * X_n$.
Докажите индукцией по n, что он обращается в 1 на $\frac{1}{3}(2^{n+1} - (-1)^{n+1})$ наборах переменных.

8. Полные системы функций и теорема Поста

Система булевых функций F называется *полной*, если формулами над этой системой можно задать любую булеву функцию.
Замыкание $[F]$ *системы функций* F — это множество всех функций, которые можно задать с помощью формул над F.
Система функций $[F]$ называется *замкнутой*, если $F = [F]$.
$\mathbf{S}_0 = \{f \mid f(0, 0, \ldots, 0) = 0\}$ — класс всех функций, сохраняющих 0.
$\mathbf{S}_1 = \{f \mid f(1, 1, \ldots, 1) = 1\}$ — класс всех функций, сохраняющих 1.
Функция $f(x_1, \ldots, x_n)$ называется *самодвойственной*, если для любого набора аргументов $(\sigma_1, \sigma_2, \ldots, \sigma_n) \in B^n$ имеет место равенство: $f(\sigma_1, \sigma_2, \ldots, \sigma_n) = \neg f(\neg\sigma_1, \neg\sigma_2, \ldots, \neg\sigma_n)$.
$\mathbf{S}$ — класс всех самодвойственных функций.
Функция $f \in \mathcal{P}_n$ называется *монотонной*, если для любых двух наборов аргументов $(\sigma_1, \sigma_2, \ldots, \sigma_n) \in B^n$ и $(\rho_1, \rho_2, \ldots, \rho_n) \in B^n$ таких, что для всех $j \in [1, n]$ $\sigma_j \geq \rho_j$, имеет место неравенство
$f(\sigma_1, \sigma_2, \ldots, \sigma_n) \geq f(\rho_1, \rho_2, \ldots, \rho_n)$.
$\mathbf{M}$ — класс всех монотонных функций.
Функция $f \in \mathcal{P}_n$ называется *линейной*, если она может быть задана линейным многочленом Жегалкина вида $\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \ldots \alpha_n X_n$, где $\alpha_i \in \{0, 1\}$ при $i = 0, 1, 2, \ldots, n$.
$\mathbf{L}$ — класс всех линейных функций.
Теорема Поста о полноте: система булевых функций F является полной тогда и только тогда, когда она не содержится ни в одном из классов $\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}, \mathbf{M}$ и $\mathbf{L}$.
Полная система булевых функций называется *базисом*, если при удалении из неё любой функции она становится неполной.

Задачи

8.1. Определите, какие из следующих формул задают несамодвойственные функции:
$A = X \vee (\neg Y \wedge Z)$,
$B = (\neg X \wedge Y) \vee (Z \wedge \neg(X + Y))$;
$C = (X \wedge \neg Z) \vee (Y \wedge \neg Z) \vee (X \wedge Y)$;
$D = (\neg X \to (Y \vee Z)) \to (\neg Y \wedge Z)$.

8.2. Определите, какие из следующих формул задают немонотонные функции:
$A = (Y \to \neg X) \to (Y \wedge Z)$;
$B = (\neg X \to (Y \wedge \neg Z)) \to Y$;

$C = \neg Z \to (Y \wedge \neg X)$.

8.3. Определите количество функций из $\mathcal{P}_n$, принадлежащих каждому из классов $\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}$ и $\mathbf{L}$.

8.4. Определите количество функций из $\mathcal{P}_n$, принадлежащих каждому из классов:
а) $\mathbf{S}_0 \cap \mathbf{S}_1$;
б) $\mathbf{S}_0 \cup \mathbf{S}_1$;
в) $\mathbf{S} \cap \mathbf{S}_0$;
г) $\mathbf{L} \cap \mathbf{S}_0$;
д) $\mathbf{L} \cap \mathbf{S}_1$;
е) $\mathbf{S} \cap \mathbf{S}_0 \cap \mathbf{S}_1$;
ж) $(\mathbf{S} \setminus \mathbf{S}_0) \cap \mathbf{S}_1$.

8.5. Доказать, что для монотонных функций $f^{(n)} \in M$ и каждого $i \in \{1, \ldots, n\}$ справедливо представление

$$f(X_1, \ldots, X_n) = (X_i \wedge f(X_1, \ldots, X_{i-1}, 1, X_{i+1}, \ldots, X_n)) \vee f(X_1, \ldots, X_{i-1}, 0, X_{i+1}, \ldots, X_n).$$

Вывести отсюда (индукцией по n), что для всякой монотонной функции, отличной от константы, существует задающая её ДНФ, не содержащая отрицаний переменных.

8.6. Сколько имеется монотонных булевых функций от n переменных, у которых значения зависят лишь от числа единиц в наборе аргументов?

8.7. Докажите, что число монотонных функций в $\mathcal{P}_n$ не меньше $2^{C_n^{\lfloor n/2 \rfloor}}$.

8.8. Найти число булевых функций от n переменных, являющихся одновременно самодвойственными и линейными.

8.9. Найти число булевых функций от n переменных, являющихся одновременно монотонными и линейными.

8.10. Доказать, что если $f(X_1, \ldots, X_n)$ — линейная функция, отличная от константы, то $|N_f^+| = 2^{n-1}$. Верно ли обратное?

8.11. Докажите полноту системы $\{\downarrow\}$, включающей только стрелку Пирса, непосредственно выразив через неё отрицание, дизъюнкцию и конъюнкцию.

8.12. Определите принадлежность каждой из функций f_1, f_2, f_3, f_4 и f_5, представленных в таблице 3, каждому из классов $\mathbf{S}_0, \mathbf{S}_1, \mathbf{S}, \mathbf{L}$ и $\mathbf{M}$.

Таблица 3. Функции f_1, f_2, f_3, f_4 и f_5

X_1	X_2	X_3	f_1	f_2	f_3	f_4	f_5
0	0	0	0	1	0	0	1
0	0	1	0	1	0	1	1
0	1	0	0	0	1	1	0
0	1	1	1	0	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	0	0	1
1	1	0	0	1	1	0	0
1	1	1	0	1	1	1	0

8.13. Используя результаты задачи 8.12, определите, какие из троек функций, представленных в таблице 3, являются полными системами. Имеются ли среди них полные системы из двух функций? Из одной функции?

8.14. Проверьте полноту системы функций $\{g, h\}$, представленных в таблице 4. Если она полна, выразите с помощью этих функций обе константы, отрицание $\neg$ и импликацию $\to$.

Таблица 4. Функции f, g и h

X_1	X_2	X_3	f	g	h
0	0	0	1	1	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	0
1	0	0	0	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	1	0

8.15. Докажите, что система $\{\vee, \wedge, \to\}$ не является полной. Можно ли её сделать полной, добавив некоторую константу?

8.16. Выразите функции $\{0, 1, \vee, \wedge, \sim\}$ с помощью формул, построенных из функций полной системы $\{\neg, \to\}$.

8.17. Какие функции следует удалить из следующей системы F, чтобы она стала базисом?
$F: f(X,Y) = X \vee Y,\ g(X,Y) = X \to \neg Y,\ h(X,Y) = X + Y$

8.18. Найдите все базисы, которые можно получить, удаляя функции из системы $\{0, 1, \wedge, \vee, \to\}$.

9. Хорновские формулы и задача получения продукции

Пусть A — это множество логических (булевых) переменных.
Хорновская (Н-) формула — это формула вида $(a_1 \wedge a_2 \wedge \ldots \wedge a_r) \to b$, где $a_i, b \in A$.
H-формула ϕ является *следствием* (или *выводится из*) множества H-формул F, если на всяком наборе значений переменных из A, на котором истинны все формулы из F, истинна и ϕ (обозначение: $F \models \phi$).
Технологический процесс t задается множеством $L_t \subseteq A$ *исходных продуктов (входов)* этого процесса и его *результирующим продуктом (выходом)* $b_t \in A$.
Задача получения продукции: выяснить по заданному набору исходных продуктов $X \subseteq A$ и результирующему продукту $y \in A$, можно ли с помощью технологических процессов из множества F получить выход y по входным продуктам из X.
Для множества технологических процессов F и исходного множества продуктов X *замыкание* X *с помощью* F — это множество продуктов $Cl(X,F) = \{y \mid (y \in A) \ \ \text{и} \ \ (F \models \bigwedge_{xinX} x \to y)\}$.

Задачи

9.1. Пусть задана система H-формул $F = \{(X \wedge Y) \to Z, (V \wedge Z) \to X, (V \wedge Z) \to Y, (Z \wedge V) \to U, (U \wedge X) \to W\}$. Какие из следующих H-формул являются следствиями системы F?
а) $(V \wedge Z) \to W$
б) $(X \wedge Y) \to W$
в) $(X \wedge Y \wedge Z) \to W$

9.2. Докажите, что последовательность процессов τ_i в доказательстве теоремы 6.1 определена корректно, т.е. все исходные продукты каждого процесса в этой последовательности имеются перед его запуском.

9.3. Докажите теорему 6.2. *Указание.* Пусть X_k — это состояние множества НОВЫЕ после k итераций основного цикла алгоритма **ЗАМЫКАНИЕ** в строках 2-6. Покажите, что для каждого продукта $z \in Cl(X, F)$, который может быть получен из X последовательностью процессов длины k, z входит в X_k.

9.4. Алгоритм **ПрямаяВолна**(X,y,F) позволяет ответить на вопрос о возможности производства y из исходных продуктов X с помощью процессов F, но в случае положительного ответа не строит последовательность процессов, приводящую к y. Измените алгоритм **ЗАМЫКАНИЕ**(X,F) так, чтобы по его результату для любого продукта $a \in Cl(X, F)$ можно было построить последовательность процессов, приводящую к a.

9.5. Назовем *сложным технологическим процессом (или производством)* такой процесс t, который по набору исходных продуктов L_t производит некоторое множество продуктов B_t (а не один продукт b_t). Обобщите алгоритм ЗАМЫКАНИЕ(X,F) так, чтобы он строил замыкание X относительно системы сложных технологических процессов F.

9.6. Используя алгоритм **ЗАМЫКАНИЕ**(X, F), вычислить замыкание для набора исходных атрибутов $X = \{c, d\}$ и следующей системы зависимостей F :

1) $a, b, d \to h$;	4) $e, f \to c$;	7) $h, d, c \to g$;
2) $a, c, d, g \to f$;	5) $b, k \to a$;	8) $d, g, a \to e$;
3) $d, g \to b$;	6) $d, c \to k$;	9) $c, d, k \to h$.

Определите, какая последовательность процессов приводит к получению e.

9.7. Пусть $A = \{a, b, c, d, e, f, g, h\}$, $X = \{b, f\}$, а множество F состоит из следующих 6 процессов:

(1) $a, b, c, h \to d$;	(3) $g, b \to e$;	(5) $f, e \to d$;
(2) $b, c, d \to a$;	(4) $e, f \to c$;	(6) $b, f \to g$.

Используя алгоритм **БыстроеЗамыкание** определите, какая последовательность процессов приводит к получению a.

9.8. Докажите теорему 6.3: *Алгоритм* **БыстроеЗамыкание***(X,F) строит замыкание* $Cl(X, F)$.

Указание. Пусть X_k — это состояние множества НОВЫЕ после k итераций основного цикла алгоритма **БыстроеЗамыкание** в строках 7-17. Покажите, что

а) если на $(k + 1)$-й итерации основного цикла для некоторого процесса t

обнаруживается, что СЧЕТ$[t] = 0$, то $L_t \subseteq X_k$;
б) для каждого продукта $z \in Cl(X, F)$, который может быть получен из X последовательностью процессов длины k, z входит в X_k;
в) условие ОБНОВА $= \varnothing$ выхода из основного цикла выполнено после $(k+1)$-ой итерации тогда и только тогда, когда $X_k = X_{k+1}$.

9.9. Измените алгоритм **БыстроеЗамыкание** так, чтобы по его результату для любого продукта $a \in Cl(X, F)$ можно было построить последовательность процессов, приводящую к a.

9.10. Используя алгоритм **БыстроеЗамыкание**, вычислить замыкание для набора исходных атрибутов $X = \{a, f\}$ и следующей системы зависимостей F :

1) $a, b, c \to h$;	3) $e, f \to c$;	5) $g, d \to e$;
2) $a, c, d, g \to h$;	4) $f, a \to d$;	6) $d, f, a \to g$.

Определите, какая последовательность процессов приводит к получению h.

10. Логика предикатов

Сигнатура языка логики предикатов: $\Sigma = \langle \mathbf{P}, \mathbf{F}, \mathbf{C} \rangle$, где $\mathbf{P} = \{P_1^{(n_1)}, P_2^{(n_2)}, \ldots, P_k^{(n_k)}, \ldots\}$ — имена предикатов, $\mathbf{F} = \{f_1^{(m_1)}, f_2^{(m_2)}, \ldots, f_j^{(m_j)}, \ldots\}$ — имена функций, $\mathbf{C} = \{c_1, c_2, \ldots\}$ — имена констант.
Var — имена *объектных переменных.*
Термы.
а) Объектная переменная из **Var** есть терм.
б) Символ константы из **C** есть терм.
в) Если $t_1, \ldots, t_k$ — термы, а $f^{(k)} \in \mathbf{F}$, то $f(t_1, \ldots, t_k)$ есть терм.
Термы, не содержащие переменных, называются *замкнутыми.*
Атомная формула.
а) Если t_1 и t_2 — термы, то выражение $(t_1 = t_2)$ является атомной формулой.
б) Любой предикатный 0-местный символ из **P** является атомной формулой.
в) Если $P^{(k)}$ $(k \geq 1)$ — предикатный k-местный символ из **P**, а $t_1, \ldots, t_k$ — термы, то выражение $P(t_1, \ldots, t_k)$ является атомной формулой.
Формула логики предикатов.
а) Всякая атомная формула есть формула.
б) Если φ — формула, то $\neg\varphi$ — формула.
в) Если φ и ψ — формулы, то выражения $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ также являются формулами.
г) Если φ — формула, а $x \in \mathbf{Var}$ — объектная переменная, то выражения

$\forall x\varphi$ и $\exists x\varphi$ являются формулами (в этом случае $\forall x\varphi$ и $\exists x\varphi$ называются областью действия*квантора всеобщности* $\forall x$ или *квантора существования* $\exists x$ соответственно).

Вхождение переменной x в формулу φ называется *связанным,* если оно входит в область действия некоторого квантора $\forall x$ или $\exists x$. В противном случае, оно называется *свободным.* Формула без свободных переменных называется *замкнутой.*

Алгебраическая система (структура, интерпретация): $\mathcal{A} = \langle A; \Sigma \rangle$, где A — основное множество системы, на котором определены интерпретации предикатов, функций и констант сигнатуры Σ.

Состояние алгебраической системы $\mathcal{A}$ — отображение $\sigma : \mathbf{Var} \to A$, которое каждой переменной $x \in \mathbf{Var}$ сопоставляет ее значение в этом состоянии $\sigma(x)$.

Значение терма $\sigma(t)$.

а) $t = x \in \mathbf{Var}$ или $t = c \in \mathbf{C}$. Тогда $\sigma(t)$ уже определено в состоянии σ.

б) $t = f(t_1, \ldots, t_m)$, где $f^{(m)} \in \mathbf{F}$, а $t_1, \ldots, t_m$ — термы. Тогда $\sigma(t)$ есть $f_A(\sigma(t_1), \ldots, \sigma(t_m))$.

Значение атомной формулы $\sigma(\varphi)$.

а) $\varphi = (t_1 = t_2)$. Тогда $\sigma(\varphi) = 1$, если $\sigma(t_1) = \sigma(t_2)$, и $\sigma(\varphi) = 0$, если $\sigma(t_1) \neq \sigma(t_2)$.

б) $\varphi = P^{(0)}$. Тогда $\sigma(P) = P_A \in \{1, 0\}$.

в) $\varphi = P(t_1, \ldots, t_k)$, где $P^{(k)} \in \mathbf{P}$, $t_1, \ldots, t_k$ — термы. Тогда $\sigma(P(t_1, \ldots, t_k)) = P_A(\sigma(t_1), \ldots, \sigma(t_k))$.

Значение формулы $\sigma(\varphi)$.

а) Если φ — атомная формула, то ее значение $\sigma(\varphi)$ определено выше.

б) Если $\varphi = \neg\varphi_1$, то $\sigma(\varphi) = \neg\sigma(\varphi_1)$.

в) Если φ имеет одну из форм $(\theta \wedge \psi)$, $(\theta \vee \psi)$ или $(\theta \to \psi)$, то $\sigma(\varphi)$ равно значению $\sigma(\theta) \wedge \sigma(\psi)$, $\sigma(\theta) \vee \sigma(\psi)$ или $\sigma(\theta) \to \sigma(\psi)$, соответственно.

г) Если $\varphi = \forall x\psi$, то $\sigma(\varphi) = 1$, если $\sigma'(\psi) = 1$ для любого состояния σ', которое может отличаться от σ только значением переменной x, т.е. такого σ', что $\sigma'(y) = \sigma(y)$ при $y \neq x$, иначе $\sigma(\varphi) = 0$.

д) Если $\varphi = \exists x\psi$, то $\sigma(\varphi) = 1$, если существует такое состояние σ', которое может отличаться от σ только значением переменной x, для которого $\sigma'(\psi) = 1$, иначе $\sigma(\varphi) = 0$.

Формула φ называется *истинной* на системе $\mathcal{A}$ ($\mathcal{A} \models \varphi$), если для любого состояния σ этой системы $\sigma(\varphi) = 1$.

Формула φ называется *тождественно истинной* (общезначимой), если она истинна на всех алгебраических системах своей сигнатуры. В этом случае пишем $\models \varphi$.

Формулы φ и ψ сигнатуры Σ называются *эквивалентными* ($\varphi \equiv \psi$), если для любой системы $\mathcal{A}$ этой сигнатуры и любого её состояния σ имеет место равенство $\sigma(\varphi) = \sigma(\psi)$.

.

Задачи

10.1. Предположим, что $P(x,y)$ означает "x — это родитель y", а $M(x)$ означает "x — это мужчина". Пусть $F(v,w) = (M(v) \wedge \exists x \exists y (P(x,y) \wedge P(x,v) \wedge \neg(y = v) \wedge P(y,w)))$. Определите, какое из следующих утверждений означает формула $F(v,w)$:

а) v — это брат w;
б) v — это племянник w;
в) v — это дядя w;
г) v — это дед w;
д) v — это двоюродный брат w.

Следующие четыре задачи относятся к системе $\mathcal{A}_1 = \langle A_1; \Sigma_1 \rangle$ сигнатуры $\Sigma_1 = \langle \mathbf{P}, \mathbf{F}, \mathbf{C} \rangle$, в которой $\mathbf{P} = \{$*живут_рядом*$^{(2)}$, *родственники*$^{(2)}$, *человек*$^{(1)}$, *число*$^{(1)}$, $\leq^{(2)}\}$, $\mathbf{F} = \{$*отец* $^{(1)}$, *лучший_друг*$^{(1)}$, *зарплата* $^{(1)}$, $+^{(2)}\}$, а $\mathbf{C} = \{$*"Пётр", "Джон", "Ольга"*, $0, 1, 2, \ldots\}$.

Её основное множество A_1 включает множество натуральных чисел $\mathbf{N} = \{0, 1, 2, \ldots\}$, для которых истинен предикат *число* и множество людей $\{$*пётр, джон, ольга, иван, мария*$\}$, для котороых истинен предикат *человек* (мы используем для объектов-людей имена, начинающиеся со строчных букв, чтобы отличить их от имён констант из сигнатуры). Пусть, кроме того, A_1 включает специальный объект ∞, соответствующий неопределенному (или ошибочному) значению. Константы интерпретируются естественным образом: числовые — соответствующими числами, имена людей — людьми с теми же именами. Функция $+$ на числах интерпретируется обычным сложением, а в остальных случаях её значением является ∞. Интерпретации остальных функций заданы в следующей таблице (для числовых аргументов все они равны ∞).

x	*отец*(x)	*лучший_друг*(x)	*зарплата*(x)
пётр	иван	джон	20
джон	пётр	ольга	35
ольга	джон	пётр	20
иван	∞	мария	40
мария	иван	пётр	30

Предикат $\leq$ интерпретируется обычным образом на числах и ложен, если хоть один его аргумент не число (мы будем для него использовать стандартную форму записи $x \leq y$ вместо $\leq (x,y)$). Остальные предикаты зададим перечислением пар, на которых они истинны:

живут_рядом $= \{$(*ольга, пётр*), (*пётр, ольга*), (*иван, джон*), (*джон,*

иван), (*мария, ольга*), (*ольга, мария*)}
родственники = {(*ольга, пётр*), (*пётр, ольга*), (*джон, пётр*), (*пётр, джон*), (*ольга, джон*), (*джон, ольга*), (*иван, мария*), (*мария, иван*)}.

10.2. Пусть $\sigma(x) =$ *иван*. Определите значения следующих формул на состоянии σ.
$\varphi_1 = \forall x(\textit{человек}(x) \to \exists y(\textit{человек}(y) \wedge \textit{живут\_рядом}(x, y))$
$\varphi_2 = \exists y(\textit{человек}(y) \wedge (\forall x \ \textit{родственники}(x, y) \vee (\neg \textit{живут\_рядом}(x, y) \wedge (35 \leq \textit{зарплата}(y)))))$

10.3. Пусть $\sigma(x) =$ *пётр*, $\sigma(y) =$ *ольга*, $\sigma(z) =$ *иван*, $\sigma(n) = 30$. Выделите в следующих формулах свободные и связанные вхождения переменных и определите значения этих формул на состоянии σ системы $\mathcal{A}_1$.
а) $\varphi_3 = \forall x \forall z(\textit{живут\_рядом}(x, z) \to \textit{родственники}(x, z))$.
б) $\varphi_4 = \forall x(\textit{живут\_рядом}(x, y) \to (\textit{отец}(x) = \textit{иван}))$.
в) $\varphi_5 = \forall u \exists v[\textit{живут\_рядом}(u, \textit{лучший\_друг}(v)) \vee (\textit{зарплата}(x) \leq \textit{зарплата}(v))]$
г) $\varphi_6 = \exists v[\neg \textit{живут\_рядом}(v, z) \to (\textit{зарплата}(\textit{отец}(v)) \leq n)]$.

10.4. Запишите формулы, выражающие следующие свойства системы $\mathcal{A}_1$ и проверьте их истинность на этой системе.
а) Рядом с каждым человеком A живет некто, отец которого является лучшим другом A.
б) У каждого человека не более одного соседа.
в) Если у соседа некоторого человека A зарплата больше чем у A, то этот сосед является родственником A.
г) Есть человек, у которого не менее 2 детей.
д) Никто не является лучшим другом для более чем 2 людей.

10.5. Запишите формулы, которые истинны на состояниях σ системы $\mathcal{A}_1$, удовлетворяющих следующим условиям.
а) $\sigma(x)$ — человек, зарплата которого больше 20 и не больше 35.
б) $\sigma(x)$ и $\sigma(y)$ — это люди, у которых один и тот же лучший друг.
в) $\sigma(x)$ — человек с максимальной зарплатой, равной $\sigma(n)$.
г) $\sigma(x)$ — человек, получающий большую зарплату, чем его отец.

Найдите все значения переменных, на которых истинны соответствующие формулы.

10.6. Арифметикой называется система $\langle \mathbf{N}; +^{(2)}, *^{(2)}; 0, 1\rangle$, где основное множество $\mathbf{N} = \{0, 1, 2, ...\}$ — множество натуральных чисел, функции $+$ и $*$ — обычное сложение и умножение, а 0 и 1 — соответствующие числа.

Запишите формулы со свободными переменными из $\{x, y, z\}$, которые истинны в арифметике тогда и только тогда, когда
а) x меньше y;
б) x является чётным числом;
в) y является простым числом;
г) x и y взаимно просты;
д) z лежит в интервале между x и y;
е) существует бесконечно много простых чисел;
ж) сложение (умножение) коммутативно.

10.7. Пусть, в системе $\mathcal{A}$ предикаты $C^{(4)}$ (*Сотрудник*(*Номер, Имя, Должность, Оклад*)), $K^{(2)}$ (*Комнаты*(*НомерСотрудника,НомерКомнаты*), $Д^{(2)}$ (*Дружат*(*Кто, С-кем*) заданы таблично следующим образом:

С(отрудник)				**К(омнаты)**		**Д(ружат)**	
№	**Имя**	**Д.**	**Окл.**	**№**	**Ком.**	**Кто**	**С кем**
1	Ваня	м.	7000	3	17	Ваня	Оля
2	Коля	эк.	6000	1	17	Маша	Оля
3	Маша	з.	6500	5	7	Галя	Ваня
4	Оля	эк.	5500	2	27	Оля	Галя
5	Галя	з.	10000	3	27	Галя	Маша

а области значений аргументов предикатов совпадают со множествами значений в соответствующих столбцах этих таблиц.(Предикат *Дружат* является симметричным и в таблице показана лишь одна половина его пар).
а) Истинна ли на $\mathcal{A}$ замкнутая формула
$\varphi = \exists x \forall y [Д(x, y) \vee \exists z [Д(x, z) \wedge \exists n \exists d \exists v \exists n_1 \exists d_1 \exists v_1 (C(n, z, d, v) \wedge C(n_1, y, d_1, v_1) \wedge$
$\exists k(K(n, k) \wedge K(n_1, k))]]$?
Ответ обоснуйте.
б) Найдите все состояния σ системы $\mathcal{A}$, на которых истинна формула
$\psi = Д(x, y) \wedge \exists n \exists d \exists m \exists d_1 (C(n, x, d, z) \wedge C(m, y, d_1, z_1) \wedge \exists k(K(n, k) \wedge K(m, k)))$.

10.8. Представьте каждое из следующих предложений формулой логики предикатов, определив в каждом случае подходящую сигнатуру.
а) Не все студенты изучают и анализ, и историю.
б) Только один студент не сдал экзамен по дискретной математике.

в) Только один студент сдал все экзамены на отлично.
г) Максимальные баллы, полученные по дискретной математике, превышают максимальные баллы, полученные по информатике.
д) Каждый, кто не любит всех вегетарианцев, является странным человеком.
е) Имеется брадобрей, который бреет тех и только тех жителей города, которые не бреются сами.
ж) Политики могут обманывать всех людей некоторое время, они могут обманывать некоторых людей всё время, но они не могут обманывать всех людей всё время.

10.9. Напишите формулу логики предикатов, которая истинна только на всех системах, основное множество которых состоит из одного элемента (2 , $3, \ldots, k$ элементов).

10.10. Докажите правило замены эквивалентных подформул:
если ψ является подформулой формулы φ, $\psi \equiv \psi_1$ и формула φ_1 получена из формулы φ заменой некоторого вхождения ψ на ψ_1, то $\varphi \equiv \varphi_1$.
Указание: используйте индукцию по построению формул и подформул.

10.11. Докажите, что бескванторная формула логики предикатов Φ, не содержащая знаков $=$, тождественно истинна тогда и только тогда, когда она может быть получена из некоторой тождественно истинной булевой формулы Ψ заменой всех вхождений каждой переменной в Ψ на некоторую атомную подформулу из Φ.

10.12. Пусть формула ψ не содержит переменную x свободно. Докажите следующие эквивалентности.
а) $\neg\forall x\varphi(x) \equiv \exists x\neg\varphi(x)$
б) $\neg\exists x\varphi(x) \equiv \forall x\neg\varphi(x)$
в) $(\psi \wedge \forall x\varphi(x)) \equiv \forall x(\psi \wedge \varphi(x))$
г) $(\psi \wedge \exists x\varphi(x)) \equiv \exists x(\psi \wedge \varphi(x))$
д) $(\psi \vee \forall x\varphi(x)) \equiv \forall x(\psi \vee \varphi(x))$
е) $(\psi \vee \exists x\varphi(x)) \equiv \exists x(\psi \vee \varphi(x))$
ж) $(\psi \to \forall x\varphi(x)) \equiv \forall x(\psi \to \varphi(x))$
з) $(\psi \to \exists x\varphi(x)) \equiv \exists x(\psi \to \varphi(x))$
и) $(\forall x\varphi(x) \to \psi) \equiv \exists x(\varphi(x) \to \psi)$
к) $(\exists x\varphi(x) \to \psi) \equiv \forall x(\varphi(x) \to \psi)$

10.13. Докажите следующие эквивалентности, показывающие, что одноимённые кванторы можно менять местами:

а) $\forall x \forall y \varphi \ \equiv \ \forall y \forall x \varphi$;
б) $\exists x \exists y \varphi \ \equiv \ \exists y \exists x \varphi$.

10.14. Докажите, что разноимённые кванторы менять местами нельзя, т.е. что формулы $\forall x \exists y \varphi$ и $\exists x \forall y \varphi$ в общем случае не эквивалентны.

10.15. Приведите к предварённой нормальной форме следующие формулы, считая, что φ и ψ — бескванторные.
а) $\neg \forall z \neg \forall x \exists y \forall u \varphi$
б) $\exists x \forall y \varphi(x, y) \to \exists x \forall y \psi(x, y)$
в) $\neg(\forall x \forall y \varphi(x, y, z) \vee (\forall z \psi(x, z) \wedge \neg \exists y \psi(x, y)))$
г) $\exists x \forall y \psi(x, y) \wedge \neg \forall x \exists y (\varphi(x) \vee \exists z \psi(y, z))$
д) $\exists x \forall y \psi(x, y) \vee \neg \exists x \exists y (\varphi(x, y) \vee \exists y \psi(y, x))$

11. Логика предикатов и базы данных

Схема отношения $R(A_1, A_2, \ldots, A_n)$ включает имя отношения R и список его атрибутов $A_1, A_2, \ldots, A_n$. Для каждого атрибута A_i определено множество $dom(A_i)$ его допустимых значений.
Схема базы данных состоит из перечня схем отношений, входящих в эту базу. В приложениях отношения чаще называют *таблицами*, их атрибуты — *столбцами*, строки таблиц — *кортежами* или *записями*, а их элементы — *полями.*
Состояние базы данных (её экземпляр) — это набор (конечных) таблиц, имеющих соответствующие схемы.
С точки зрения логики предикатов состояние базы данных — не что иное, как некоторая конечная система сигнатуры, включающей предикаты, соответствующие отношениям её схемы, одноместные предикаты для каждой из областей атрибутов и стандартные отношения, определённые на этих областях (равенство, порядок и др.)

Реляционная алгебра.
Реляционные операторы.
I) *Теоретико-множественные операции*: объединение, пересечение, вычитание и декартово произведение.
II) *Специальные реляционные операторы.*
Оператор выбора, применённый к отношению $R(A_1, \ldots, A_n)$:
$P = \sigma_C(R) = \{(a_1, \ldots, a_n) \mid a_i \in dom(A_i)(i = 1, \ldots, n) \text{ и } C(a_1, \ldots, a_n) = 1\}$,

где $C(A_1, \ldots, A_n)$ — булева формула, которая построена из элементарных условий (равенств, неравенств), включающих имена атрибутов R и константы.

Оператор проекции отношения $R(A_1, \ldots, A_n)$ на подмножество атрибутов $X = \{A_{i_1}, \ldots, A_{i_k}\}$ $P = \pi_X(R)$ возвращает в P все кортежи R, из которых удалены значения атрибутов, не входящих в X.

Оператор естественного соединения $P = R \bowtie S$ отношений $R(A_1, \ldots, A_n, B_1, \ldots, B_m)$ и $S(B_1, \ldots, B_m, C_1, \ldots, C_k)$ с общими атрибутами $B_1, \ldots, B_m$ возвращает отношение со схемой $P(A_1, \ldots, A_n, B_1, \ldots, B_m, C_1, \ldots, C_k)$, которое содержит кортежи, составленные из кортежей отношения R, продолженных кортежами отношения S. При этом соединяются лишь пары кортежей $r \in R$ и $s \in S$, имеющих одинаковые значения всех общих атрибутов $B_1, \ldots, B_m$.

Оператор тета-соединения $P_1 = R \bowtie_C S$ отношений R и S возвращает кортежи, которые составлены из кортежей отношения R, продолженных кортежами отношения S, удовлетворяющими условию C. Синтаксис этого условия такой же, как и у оператора выбора.

Атрибуты R и S с одинаковыми именами входят в схему P_1 дважды (обычно, как и в случае декартова произведения, перед ними помещается через точку имя отношения).

Задачи

11.1. Пусть $dom(A) = \{a_1, a_2, a_3\}, dom(B) = dom(D) = \mathbf{N}, dom(C) = \{c_1, c_2, c_3\}$. Пусть отношение r, со схемой $R(A, B, C)$, включает кортежи $\{(a_1, 7, c_1), (a_1, 5, c_2), (a_2, 7, c_1), (a_3, 4, c_3)\}$, а отношение s, имеющее схему $S(C, D)$, включает кортежи $\{(c_1, 2), (c_2, 8), (c_2, 7), (c_1, 4), (c_2, 5)\}$.

Определите результаты следующих выражений реляционной алгебры:

а) $\sigma_{A=a_1} r$;

б) $\pi_B r \times \pi_D s$;

в) $r \bowtie s$;

г) $\pi_{A,C} r \bowtie \pi_C r$;

д) $r \bowtie_{r.C \neq s.C \wedge r.B > s.D} s$.

11.2. Какие из следующих равенств выражений реляционной алгебры верны для любых отношений со схемами $R(A, B, C)$ и $S(A, B, C)$?

а) $\sigma_{A=a}(\sigma_{B>b}(R - S)) = \sigma_{B>b}(\sigma_{A=a}(R - S))$,

б) $\pi_{BA}(\sigma_{A=a}(R)) = \sigma_{A=a}(\pi_{BA}(R))$,

в) $\pi_{BC}(R \cap S) = \pi_{BC}(R) \cap \pi_{BC}(S)$.

Сотрудники

Номер	ФИО	Отдел	Должность	Оклад
1	Иванов А. А.	торговый	менеджер	7000
2	Сидоров Н. П.	плановый	экономист	5000
3	Сидорова М. И.	торговый	зав. складом	6000
4	Ольгина Н. А.	плановый	экономист	5500
5	Горев С. В.	плановый	зав. отделом	10000

Комнаты

НомерСотрудника	Этаж	НомерКомнаты
3	2	17
1	2	17
7	2	18
5	3	7
2	3	27

11.3. Напишите SQL-запросы и соответствующие формулы для получения информации из приведенной выше базы данных (*Сотрудники, Комнаты*) и получите ответы на эти запросы.
а) Найти всех сотрудников с окладом больше 5500.
б) Составить список должностей и получаемых по ним окладов.
в) Найти все отделы, в которых есть сотрудники с окладом $>$ 8000.
г) Составить список сотрудников торгового отдела, получающих зарплату от 6000 до 6500 и работающих не на 3-м этаже.
д) Составить список комнат, в которых есть сотрудники с окладом меньше 5500 или больше 7500.

11.4. Определите, какие из приведенных ограничений целостности Φ_1, Φ_2, Φ_3 выполняются для приведенного выше состояния базы данных (*Сотрудники, Комнаты*).
$\Phi_1 = \forall \textit{Номер} \forall x \forall y \forall z \forall v \forall x_1 \forall y_1 \forall z_1 \forall v_1 [(\textit{Сотрудники}(\textit{Номер}, x, y, z, v) \wedge \textit{Сотрудники}(\textit{Номер}, x_1, y_1, z_1, v_1)) \rightarrow ((x = x_1) \wedge (y = y_1) \wedge (z = z_1) \wedge (v = v_1))]$
$\Phi_2 = \forall \textit{НомерСотрудника} \forall x \forall y [\textit{Комнаты}(\textit{НомерСотрудника}, x, y) \rightarrow \exists z \exists u \exists v \exists w (\textit{Сотрудники}(\textit{НомерСотрудника}, z, u, v, w))]$
$\Phi_3 = \forall x \forall y \forall z \forall v \forall \textit{Оклад}(\textit{Сотрудники}(x, y, z, v, \textit{Оклад}) \rightarrow ((1000 < \textit{Оклад}) \wedge (\textit{Оклад} < 25000)))$

11.5. Напишите формулы, выражающие следующие ограничения целостности для базы (*Сотрудники, Комнаты*), и определите, какие из них вы-

полняются для приведенного выше её состояния.
а) В таблице *Комнаты* набор полей (*НомерСотрудника, НомерКомнаты*) является ключом.
б) Для каждого сотрудника из таблицы *Сотрудники* в таблице *Комнаты* определено его место работы.
в) Номера всех комнат на 2-м этаже больше 10, но меньше 20, а номера всех комнат на 3-м этаже больше 20.

12. Графы: представления, достижимость и связность

Ориентированный граф G — это пара (V, E), где V — конечное множество *вершин* (узлов, точек) графа, а $E \subseteq V \times V$ — некоторое множество пар вершин, или бинарное отношение на V. Элементы E называют *рёбрами*.
Неориентированный граф $G = (V, E)$ — это ориентированный граф, у которого для каждого ребра $(u, v) \in E$ имеется противоположное ребро $(v, u) \in E$, т.е. отношение E симметрично.
Если $e = (u, v) \in E$, то вершины u и v называются *смежными* в G, а ребро e и эти вершины называются *инцидентными*.
В ориентированном графе может быть ребро вида (u, u), называемое *петлёй*, а в неориентированном петель не бывает.
Степенью вершины в неориентированном графе называется число смежных с ней вершин.
В ориентированном графе *полустепень исхода* вершины — это число исходящих из неё рёбер, а *полустепень захода* — это число входящих в данную вершину рёбер.

Размеченный граф — это ориентированный или неориентированный граф $G = (V, E)$, снабжённый одной или двумя функциями разметки вида: $l : V \to M$ и $c : E \to L$, где M и L — множества меток вершин и рёбер, соответственно.
Мультиграф отличается от графа тем, что в нем между двумя вершинами может быть несколько ребер, которые обычно различаются своими метками.
Два графа $G_1 = (V_1, E_1)$ и $G_2 = (V_2, E_2)$ называются *изоморфными*, если между их вершинами существует взаимно однозначное соответствие $\varphi : V_1 \to V_2$ такое, что для любой пары вершин u, v из V_1 ребро $(u, v) \in E_1$ тогда и только тогда, когда ребро $(\varphi(u), \varphi(v)) \in E_2$.
Путь в ориентированном или неориентированном графе — это последовательность рёбер вида $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$. Этот путь ведет из начальной вершины v_1 в конечную вершину v_n и имеет длину $n - 1$. В этом

случае будем говорить, что v_n *достижима* из v_1. Будем считать, что каждая вершина достижима сама из себя путём длины 0.

Путь можно также определять как соответствующую последовательность вершин: $(v_1, v_2, v_3, \ldots, v_{n-1}, v_n)$, где $(v_i, v_{i+1}) \in E$ при $i = 1, 2, \ldots, n-1$.

Путь назывется *простым*, если все ребра и все вершины на нём, кроме, быть может, первой и последней, различны.

Циклом в ориентированном графе называется путь, в котором начальная вершина совпадает с конечной и который содержит хотя бы одно ребро. Цикл $(v_1, v_2, \ldots, v_{n-1}, v_n = v_1)$ называется *простым*, если в нём нет одинаковых вершин, кроме первой и последней, т.е. если все вершины $v_1, \ldots, v_{n-1}$ различны.

В неориентированном графе путь $(v_1, v_2, \ldots, v_{n-1}, v_n = v_1)$ называется *циклом*, если $n \geq 4$ и все рёбра (v_i, v_{i+1}) различны.

Если в графе нет циклов, то он называется *ациклическим*.

Неориентированный граф называется *связным*, если любая пара вершин в нём соединена путём. При выполнении такого же условия ориентированный граф называется *сильно связным*.

Матрица смежности ориентированного (или неориентированного) графа $G = (V, E)$ с n вершинами $V = \{v_1, \ldots, v_n\}$ — это булева матрица A_G размера $n \times n$ с элементами

$$a_{ij} = \begin{cases} 1, & \textit{если } \ (v_i, v_j) \in E \\ 0 & \textit{в противном случае} \end{cases}$$

Матрица инцидентности ориентированного (или неориентированного) графа $G = (V, E)$ с n вершинами $V = \{v_1, \ldots, v_n\}$ и m рёбрами $E = \{e_1, \ldots, e_m\}$ называется матрица B_G размера $n \times m$ с элементами

$$b_{ij} = \begin{cases} 1, & \textit{если для некоторого } k \ \textit{ ребро } \ e_j = (v_i, v_k), \\ -1, & \textit{если для некоторого } k \ \textit{ ребро } \ e_j = (v_k, v_i), \\ 2, & \textit{если ребро } \ e_j = (v_i, v_i), \\ 0 & \textit{в противном случае} \end{cases}$$

Список смежности L_v для вершины v включает все смежные с ней вершины, т.е. $L_v = w_1, \ldots, w_k$, где $\{w_1, \ldots, w_k\} = \{w \mid (v, w) \in E\}$.

Представление $G = (V, E)$ с n вершинами $V = \{v_1, \ldots, v_n\}$ с помощью *списков смежности* состоит из списков смежности всех вершин: $L_{v_1}, L_{v_2}, \ldots, L_{v_n}$.

Вершина w *достижима* из вершины v в графе $G = (V, E)$, если $v = w$ или в G есть путь из v в w.

Связные компоненты в неориентированном графе — это классы эквивалентности по отношению достижимости.

Вершины v и w ориентированного графа $G = (V, E)$ называются *взаимно достижимыми*, если в G есть путь из v в w и путь из w в v.

Классы эквивалентности по отношению взаимной достижимости называются *компонентами сильной связности*, или *двусвязными компонентами*, графа.
Граф достижимости $G^* = (V, E^*)$ для ориентированного графа $G = (V, E)$ имеет то же множество вершин V и множество ребер $E^* = \{(u, v) \mid$ в графе G вершина v достижима из вершины $u\}$.
Граф сильной достижимости $G^*_* = (V, E^*_*)$ для G ориентированного графа $G = (V, E)$ имеет то же множество вершин V и множество ребер $E^*_* = \{(u, v) \mid$ в графе G вершины v и u взаимно достижимы$\}$.

Пусть K и K' — компоненты сильной связности графа G. Компонента K *достижима из* компоненты K', если $K = K'$ или существуют такие две вершины $u \in K$ и $v \in K'$, что u достижима из v. K *строго достижима из* K', если $K \neq K'$ и K достижима из K'.
Компонента K называется *минимальной,* если она не является строго достижимой ни из какой компоненты.
Подмножество вершин $W \subseteq V$ ориентированного графа $G = (V, E)$ называется *порождающим*, если из вершин W можно достичь любую вершину графа.
Подмножество вершин $W \subseteq V$ называется *базой графа*, если оно является порождающим, но никакое его собственное подмножество порождающим не является.
Теорема. Подмножество вершин W является базой G тогда и только тогда, когда оно содержит по одной вершине из каждой минимальной компоненты сильной связности G и не содержит никаких других вершин.

Задачи

12.1. Постройте представления в виде матрицы смежности, матрицы инцидентности и списков смежности для ориентированного графа $G = (V, E)$, где $V = \{a, b, c, d\}, E = \{(a, b), (a, c), (a, a), (b, a), (b, b), (c, a), (c, d), (d, b)\}$.

12.2. Неориентированный граф называется полным, если для каждой пары разных вершин имеется соединяющее их ребро. Найдите число рёбер в полном n-вершинном графе.

12.3. Пусть $G = (V, E)$ — это ориентированный граф без циклов и $|E| > 0$. Какие из следующих утверждений верны?
а) В G есть вершина, в которую не входят ребра.
б) В G есть вершина, из которой не выходят ребра.
в) В G есть изолированная вершина, т.е. вершина, у которой нет инцидентных рёбер.

12.4. Докажите, что сумма степеней всех вершин произвольного неориентированного графа равна удвоенному числу рёбер.

У этой задачи имеется популярная интерпретация: доказать, что общее число рукопожатий, которыми обменялись люди, пришедшие на вечеринку, всегда чётно (*Лемма о рукопожатиях*).

12.5. Может ли в государстве, в котором из каждого города выходит ровно пять дорог, быть ровно 77 дорог между городами? А 80 дорог?

12.6. а) Докажите, что если неориентированный граф $G = (V, E)$ не является связным графом, то его дополнение – граф $\overline{G} = (V, \overline{E})$ является связным (здесь $\overline{E} = \{(u, v) \mid u \in V, v \in V$ и $(u, v) \notin E\}$).
б) У задачи из п. (а) имеется следующая популярная интерпретация. В стране Приозерия каждая пара городов соединена только одним транспортным маршрутом: или водным, или автобусным. Доказать, что существует вид транспорта, которым можно доехать из любого города страны в любой другой (возможно с пересадками).

12.7. Перечислите все неизоморфные неориентированные графы, у которых не более четырёх вершин.

12.8. Докажите, что неориентированный связный граф остается связным после удаления некоторого ребра тогда и только тогда, когда это ребро принадлежит некоторому циклу.

12.9. Докажите, что неориентированный связный граф с n вершинами
а) содержит не менее $n - 1$ рёбер,
б) если содержит больше $n - 1$ рёбер, то имеет хотя бы один цикл.

12.10. Докажите, что в любой группе из 6 человек есть трое попарно знакомых или трое попарно незнакомых.

12.11. Докажите, что неориентированный граф $G = (V, E)$ связен тогда и только тогда, когда для каждого разбиения $V = V_1 \cup V_2$ с непустыми V_1 и V_2 существует ребро, соединяющее V_1 с V_2.

12.12. Докажите, что, если в неориентированном графе имеется ровно две вершины нечётной степени, то они связаны путём.

12.13. Докажите, что в связном неориентированном графе любые два простых пути максимальной длины имеют общую вершину.

12.14. Пусть неориентированный граф без петель $G = (V, E)$ имеет k компонент связности. Доказать, что тогда
$$|E| \leq (|V| - k)(|V| - k + 1)/2.$$

12.15. Определите, что представляет собой граф достижимости для
а) графа с n вершинами и пустым множеством рёбер;
б) графа с n вершинами: $V = \{v_1, \ldots, v_n\}$, рёбра которого образуют цикл: $E = \{(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n), (v_n, v_1)\}$.

12.16. Пусть граф $G = (V, E)$ задан своей матрицей смежности

$$A_G = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Постройте граф достижимости $G^* = (V, E^*)$ для G и определите, сколько в нём новых рёбер, т.е. чему равна разность $|E^*| - |E|$.

12.17. Чему равно число связных компонент неориентированного графа $G = (V, E)$, где $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, E = \{(1, 4), (2, 7), (3, 9), (7, 4), (1, 5), (6, 7)\}$?

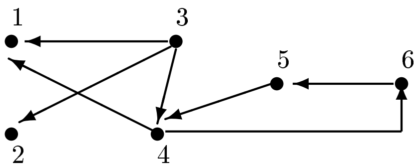


Рис. 1. Граф G_1

12.18. Построить для заданного на рис. 1 ориентированного графа $G_1 = (V, E)$ его матрицу смежности A_{G_1}, матрицу инцидентности B_{G_1} и списки смежности. Вычислить матрицу достижимости $A_{G_1^*}$ и построить соответствующий граф достижимости G_1^*.

12.19. Вычислите матрицу графа достижимости A_{G^*} для графа
$G = (\{a, b, c, d, e, f, g\}, \{(a, b), (b, a), (a, c), (b, d), (e, d), (d, f), (f, c), (c, f),$

$(g, e)\}$
и постройте соответствующий ей граф достижимости. Найдите все базы графа G.

12.20. Определить для каждого из следующих ориентированных графов G_i $(i = 1, 2, 3)$, изображённых на рис. 2, 3 и 4, компоненты сильной связности, порядок (отношение достижимости) на них и все базы.

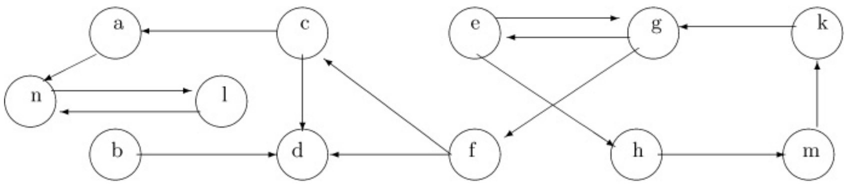


Рис. 2. Граф G_1

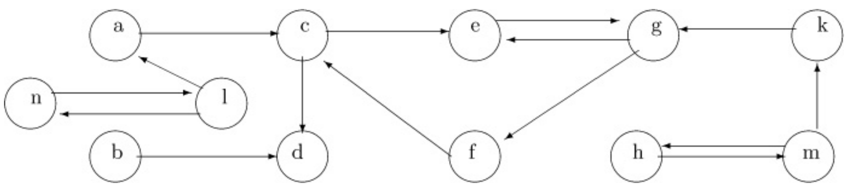


Рис. 3. Граф G_2

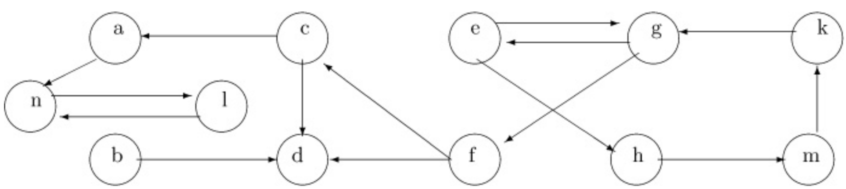


Рис. 4. Граф G_3

12.21. Доказать, что в ориентированном графе без циклов существует единственная база, состоящая из всех вершин с нулевой степенью захода.

13. Деревья

(1) Неориентированный граф называется *деревом*, если он связный и в нём нет циклов.

Эквивалентные определения неориентированного дерева.

(2) Для любых двух вершин в G есть единственный соединяющий их путь.
(3) G связен, но при удалении из E любого ребра перестает быть связным.
(4) G связен и $|E| = |V| - 1$.
(5) G ациклический и $|E| = |V| - 1$.
(6) G ациклический, но добавление любого ребра к E порождает цикл.

Ориентированный граф $G = (V, E)$ называется *(ориентированным) деревом*, если
1) в нём есть одна вершина $r \in E$, в которую не входят рёбра; она называется *корнем* дерева;
2) в каждую из остальных вершин входит ровно по одному ребру;
3) все вершины достижимы из корня.
Индуктивное определение класса ориентированных деревьев $\mathcal{D}$.
1) Граф $T_0 = (V, E)$, с единственной вершиной $V = \{v\}$ и пустым множеством ребер $E = \varnothing$ является деревом (входит в $\mathcal{D}$). Вершина v называется корнем этого дерева.
2) Пусть графы $T_1 = (V_1, E_1), \dots, T_k = (V_k, E_k)$ с корнями $r_1 \in V_1, \dots, r_k \in V_k$ принадлежат $\mathcal{D}$, а r_0 — новая вершина, т.е. $r_0 \notin \bigcup_{i=1}^{k} V_i$. Тогда классу $\mathcal{D}$ принадлежит также следующий граф $T = (V, E)$, где $V = \{r_0\} \cup \bigcup_{i=1}^{k} V_i$, $E = \{(r_0, r_i) \mid i = 1, \dots, k\} \cup \bigcup_{i=1}^{k} E_i$.
Корнем этого дерева является вершина r_0.
3) Других графов в классе $\mathcal{D}$ нет.

Листья — это вершины, из которых не выходят рёбра.
Путь из корня в лист называется *ветвью* дерева.
Высота дерева — это максимальная из длин его ветвей.
Глубина вершины — это длина пути из корня в эту вершину.
Высота вершины v — это максимальная из длин путей из v в лист.
Граф, являющийся объединением нескольких непересекающихся деревьев, называется *лесом*.
Пусть $T = (V, E)$ — ориентированное дерево.
Если $(v, w) \in E$, то v называется *отцом* или *родителем* w, а w — *сыном* или *ребёнком* v.
Если из вершины v ведет путь в вершину w, то v называется *предком* w, а w — *потомком* v. Вершины, у которых общий отец, называются *братьями* или *сёстрами*.

Ориентированное дерево называется *бинарным* или *двоичным*, если у каждой его внутренней вершины имеется не более двух сыновей, причем рёбра, ведущие к ним, помечены двумя разными метками (обычно используются метки из пар: "левый" – "правый", 0 – 1, + – − и т.п.)
Бинарное дерево называется *полным*, если все его ветви имеют одинаковую длину.

Прямой (префиксный) обход дерева основан на принципе: "сначала родитель, затем дети". Определим индукцией по построению дерева T его прямое представление $\mathit{ПР}(T)$:
1) Если $T_0 = (\{v\}, \varnothing)$, то $\mathit{ПР}(T_0) = v$.
2) Если T получено из деревьев $T_1, \ldots, T_k$ и нового корня r_0 по пункту (2) индуктивного определения дерева, то $\mathit{ПР}(T) = r_0\, \mathit{ПР}(T_1) \ldots \mathit{ПР}(T_k)$.
Обратный (суффиксный) обход дерева основан на противоположном принципе: "сначала дети , затем родитель". Его индуктивное определение:
1) Если $T_0 = (\{v\}, \varnothing)$, то $\mathit{ОБР}(T_0) = v$.
2) Если T получено из деревьев $T_1, \ldots, T_k$ и нового корня r_0 по пункту (2) индуктивного определения дерева, то $\mathit{ОБР}(T) = \mathit{ОБР}(T_1) \ldots \mathit{ОБР}(T_k)\, r_0$.
Инфиксный (внутренний) обход бинарного дерева основан на принципе: "сначала левый сын, затем родитель, а затем правый сын".
1) Если $T_0 = (\{v\}, \varnothing)$, то $\mathit{ИНФ}(T_0) = v$.
2) Если T получено из деревьев T_1, T_2 и нового корня r_0 по пункту (2) индуктивного определения дерева, то $\mathit{ИНФ}(T) = \mathit{ИНФ}(T_1) r_0 \mathit{ИНФ}(T_2)$
(Если одно из деревьев T_1, T_2 пусто, то соответствующее ему инфиксное представление тоже пусто).

Задачи

13.1. Докажите, что если в неориентированном дереве имеется ровно две вершины степени 1, то оно является путём.

13.2. Центром неориентированного графа G называется вершина v, для которой длина максимального пути от неё до остальных вершин минимальна. Докажите, что в дереве может быть один центр или два смежных центра.

13.3. Приведите пример ориентированного графа, для которого выполнены условия 1) и 2) из определения дерева, но который не является деревом.

13.4. Некоторый слух распространялся следующим образом: его источник позвонил трём своим друзьям, каждый из них передал по телефону этот слух четырём своим друзям, а каждый из них, в свою очередь, передал его пяти своим друзьям. Сколько человек узнали слух, если никто из них не получил более одного звонка и никто не звонил источнику слуха? Сколько звонков было произведено?

13.5. Пусть корень ориентированного дерева T имеет 3 сыновей, а каждая из остальных внутренних вершин имеет два или четыре сына, при этом число вершин с 2 сыновьями вдвое меньше числа вершин с 4. Сколько всего вершин в T, если известно, что число его листьев равно 38?

13.6. Пусть корень ориентированного дерева T имеет 5 сыновей, а каждая из остальных внутренних вершин имеет три или четыре сына, при этом число вершин с 3 сыновьями вдвое больше числа вершин с 4. Сколько всего вершин и рёбер в T, если известно, что число его листьев равно 26?

13.7. Докажите, что если в связном неориентированном графе число вершин равно числу рёбер, то можно выбросить одно из рёбер так, что после этого граф станет деревом.

13.8. Пусть $G = (V, E)$ — неориентированное дерево и $v \in V$ — произвольная вершина. Докажите, что если для каждого ребра $(u, w) \in E$ выбрать ориентацию от u к w, если им заканчивается путь из v в w, и ориентацию от w к u, если им заканчивается путь из v в u, то полученный ориентированный граф будет ориентированным деревом с корнем v.

Используйте это утверждение для доказательства следующего факта: если в неориентированном дереве $G = (V, E)$ имеется вершина степени $d > 1$, то в нем имеется по крайней мере d вершин степени 1.

13.9. Пусть $T = (V, E)$ — это ориентированное дерево с корнем $v_0 \in V$. Определим для каждой вершины $v \in V$ подграф $T_v = (V_v, E_v)$ следующим образом: V_v — это множество вершин, достижимых из v в T, а E_v — это множество ребер из E, оба конца которых входят в V_v. Доказать, что
а) T_v является деревом с корнем v;
б) если две разные вершины v и u имеют одинаковую глубину, то деревья T_v и T_u не пересекаются.

13.10. Пусть $G = (V, E)$ — ориентированный граф с $n > 1$ вершинами. Докажите, что G является (ориентированным) деревом тогда и только тогда, когда в G нет циклов, имеется одна вершина r, в которую не входят ребра, а в каждую из остальных вершин $v \in V \setminus \{r\}$ входит ровно одно ребро.

13.11. Для каждого из обходов деревьев *ПР*(T), *ОБР*(T) и *ИНФ*(T) предложите процедуру, восстановления соответствующего дерева $T \in \mathcal{T}(\mathbf{F}, \mathbf{C}, \mathbf{Var})$.

13.12. Докажите по индукции, что в любом бинарном дереве число вершин степени 2 на единицу меньше числа листьев.

13.13. Определите число листьев и число вершин в полном бинарном дереве высоты h.

13.14. Определите прямой, обратный и инфиксный обходы дерева T_1, изображенного на рис. 5.

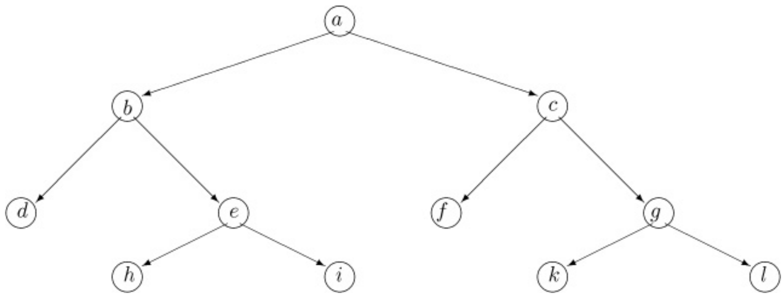


Рис. 5. Дерево T_1

13.15. Определите прямой и обратный обходы дерева T_2, изображенного на рис. 6.

13.16. Постройте дерево, представляющее следующую логическую формулу
$\Psi = ((X \vee \neg Y) \wedge \neg(Z \to (X \wedge Y))) \vee (\neg Z + Y)$.
Для полученного дерева определите прямой, обратный и инфиксный обходы.

13.17. Пусть $xzuv/-*yz-xy+*+$ — это обратный обход дерева арифметической формулы, составленной из переменных x, y, z, u, v и знаков операций $+, -, *, /$. Восстановите это дерево и формулу.

13.18. Постройте дерево и ациклический ориентированный граф, представляющие следующую арифметическую формулу:
$\Phi = (a+b)/(c+a*d) + ((c+a*d) - (a+b)*(c-d))$.

Сколько вершин удалось сократить?

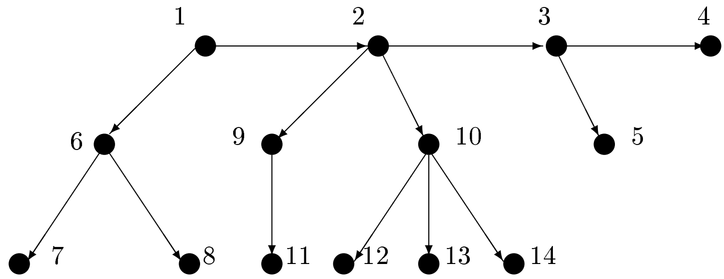


Рис. 6. Дерево T_2

14. Алгоритмы на графах

Цикл в связном неориентированном графе называется *эйлеровым,* если он проходит по одному разу через каждое ребро графа.

Неориентированный граф $G = (V, E)$ называется *двудольным,* если его вершины можно разбить на две непересекающихся части X и Y ($V = X \cup Y, X \cap Y = \varnothing$) так, что каждое ребро из $e \in E$ соединяет вершину из X с вершиной из Y. Такой граф также называется *бихроматическим,* так как его вершины можно раскрасить в два цвета так, что соседние вершины будут окрашены в разные цвета.

Граф $G_1 = (V_1, E_1)$ называется *подграфом* графа $G = (V, E)$, если $V_1 \subseteq V$ и $E_1 \subseteq E$.
Остовом (остовным деревом, каркасом , скелетом) (неориентированного) связного графа $G = (V, E)$ называется его подграф $S = (V, T)$, являющийся деревом.
Пусть задана функция $c : E \to \mathbf{R}$, приписывающая каждому ребру $e \in E$ его стоимость (вес, длину) $c(e) \in \mathbf{R}$ ($\mathbf{R}$ — множество вещественных чисел). Тогда *стоимость* $c(S)$ дерева S — это сумма стоимостей всех его рёбер, т.е. $c(S) = \sum\limits_{e \in T} c(e)$.
Минимальным остовом называется остов минимальной стоимости.

Дерево $S = (V, T)$, которое строится алгоритмом поиска в глубину **ПОГ**, называется *глубинным остовом* или *глубинным остовным деревом* графа G. Рёбра, попавшие в множество T, называются *прямыми*, а не попавшие в это множество рёбра из множества $(E \setminus T)$ — *обратными.*

Ребро (v, w) неориентированного графа $G = (V, E)$ называется *мостом* G, если при его удалении из E число связных компонент графа увеличивается.

Пусть $NUM[w]$ — номер, который алгоритм поиска в глубину **ПОГ** присваивает вершине w.
ВЕРХ(w) это минимум из $NUM[w]$ и наименьшего из номеров вершин, к которым ведут обратные рёбра от w и её потомков.
Ребро (v, w) глубинного остова $D = (V, E)$ является мостом неориентированного графа $G = (V, E)$ тогда и только тогда, когда ВЕРХ$(w) = NUM[w]$.

Пусть $G = (V, E)$ — ориентированный граф, $c(e) \geq 0$ — длина (вес, стоимость) ребра $e \in E$. Тогда длина пути $p = v_1, v_2, \ldots, v_{k+1}$ — это сумма длин рёбер, входящих в этот путь: $c(p) = \sum\limits_{i=1}^{k} c(v_i, v_{i+1})$. Если в G имеется путь из вершины a в вершину b, то имеется и такой путь минимальной длины. Он называется *кратчайшим путем из a в b*.

Задачи

14.1. Докажите, что в в связном неориентированном графе имеется эйлеров цикл тогда и только тогда, когда степени всех его вершин чётны (такие графы называются *чётными*).
Указание. Достаточность можно установить, доказав правильность следующей процедуры построения эйлерова цикла.
1) Выбрать произвольную вершину a и построить цикл, начинающийся и закачивающийся в a, следуя правилу $(*)$: *прийдя в некоторую вершину, выйти из неё по произвольному ребру, ещё не включённому в цикл.*
2) Если построенный цикл содержит не все рёбра, то выбрать среди вершин, через которые он проходит, произвольную вершину b, инцидентную ещё не пройденному ребру (почему такая вершина найдётся?), и построить, начиная с неё, цикл, следуя тому же правилу $(*)$. Объединить этот цикл с построенным ранее.
3) Повторять пункт (2) до тех пор, пока построенный цикл не станет эйлеровым.

14.2. Используя результаты предыдущей задачи, определить по неориентированному графу $G = (V, E)$, чётный ли он. Если он не является чётным, то удалить из него минимальное число рёбер, чтобы он стал чётным. Построить в исходном или в получившемся после удаления рёбер чётном графе Эйлеров цикл.

$V = \{a, b, c, e, f, g, h, k, m, n\}$, $E = \{(a, c), (a, h), (a, m), (a, k), (b, c), (b, k), (b, f), (b, m), (c, k), (c, m), (e, f), (e, g), (f, k), (f, n), (g, m), (g, h), (h, k), (h, m), (k, n)\}$.

14.3. Докажите, что граф является двудольным тогда и только тогда, когда в нем нет циклов нечётной длины (*теорема Кёнига*).

Указание. Достаточность можно установить, доказав правильность следующей процедуры разбиения V на X и Y:

1) Поместить произвольную вершину $v \in V$ в X и отметить ее знаком $+$.
2) ПОКА имеются неотмеченные вершины с отмеченными соседями
ВЫПОЛНЯТЬ {
2.1) Поместить все неотмеченные вершины с соседями из X в Y и отметить их знаком $-$;
2.2) Поместить все неотмеченные вершины с соседями из Y в X и отметить их знаком $+$;
};
3) ЕСЛИ после завершения цикла 2 в V остались неотмеченные вершины
ТО поместить произвольную такую вершину v в X, отметить ее знаком $+$ и снова повторить
цикл 2
ИНАЧЕ выдать в качестве результата полученные множества: X – вершины, отмеченные $+$,
и Y – вершины, отмеченные $-$.

Для доказательства корректности этой процедуры установите, что в процессе разметки ни одна вершина не получит соседа с той же меткой.

14.4. Используя результаты предыдущей задачи, определить, является ли заданный ниже неориентированный граф $G = (V, E)$ двудольным. Если он не двудольный, то каково минимальное число рёбер, которые нужно из него удалить, чтобы он стал двудольным? Приведите обоснование ответа.
$V = \{a, b, c, e, f, g, h, k, m, n\}$, $E = \{(a, h), (a, n), (a, k), (b, k), (b, f), (b, m), (c, k), (c, h), (e, f), (e, g), (f, a), (f, m), (g, m), (m, n)\}$.

14.5. Найти минимальное остовное дерево для неориентированного графа $G = (V, E)$, где $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, $E = \{(v_1, v_2, 18), (v_1, v_3, 2), (v_3, v_2, 4), (v_3, v_4, 6), (v_3, v_5, 8), (v_4, v_6, 5), (v_5, v_4, 4), (v_6, v_1, 7), (v_6, v_8, 4), (v_6, v_7, 3), (v_7, v_5, 1), (v_7, v_8, 7), (v_8, v_1, 5), (v_8, v_9, 3), (v_9, v_1, 1)\}$ (третий параметр в скобках — стоимость ребра).

14.6. Найти минимальное остовное дерево для неориентированного графа $G = (V, E)$, где $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, $E = \{(v_1, v_2, 10)$, $(v_1, v_4, 6), (v_2, v_3, 12), (v_2, v_4, 3), (v_3, v_5, 20), (v_3, v_6, 11), (v_4, v_5, 3), (v_4, v_7, 2)$, $(v_5, v_6, 4), (v_5, v_7, 2), (v_5, v_8, 5), (v_5, v_9, 9), \quad (v_6, v_8, 6), (v_6, v_9, 17), (v_8, v_9, 7)\}$ (третий параметр в скобках — стоимость ребра).

14.7. Пусть задан неориентированный нагруженный граф $G:$ $V = \{a, b, c, d, e, f, g, h, k\}$, $E = \{(a, b; 10), (a, c; 7), (b, f; 21), (b, d; 9), (c, d; 8)$, $(f, e; 7), (f, g; 8), (e, k; 12), (e, h; 10), (g, h; 8)\}$ (здесь каждая скобка $(u, v; D)$ задает ребро $(u, v) \in E$ и его "вес" $c(u, v) = D$). Какие из рёбер (a, b), (e, h), (b, f) не могут попасть ни в какой минимальный остов?

14.8. Пусть задан неориентированный граф $G = (V, E):$ $V = \{a, b, c, d, e, f, g, h, i\}, E = \{(a, b), (a, c), (b, d), (b, c), (d, e), (d, f), (f, g)$, $(f, h), (f, i)\}$. Используя вариант поиска в глубину с подсчетом функции ВЕРХ, определите все мосты этого графа и укажите их число.

14.9. Пусть $S = (V, T)$ — остовное дерево наименьшей стоимости, построенное алгоритмом МИНОД для нагруженного неориентированного графа $G = (V, E)$ с n вершинами. Пусть $c_1 \le c_2 \le \ldots \le c_{n-1}$ — это последовательность длин ребер из T, упорядоченных по возрастанию. Пусть S' — произвольное остовное дерево для G с длинами рёбер $d_1 \le d_2 \le \ldots \le d_{n-1}$. Показать, что $c_i \le d_i$ для всех $i : 1 \le i \le n - 1$.

14.10. Пусть e — ребро максимального веса в некотором цикле графа $G = (V, E)$. Докажите, что существует минимальный остов графа $G' = (V, E \setminus \{e\})$, который является также минимальным остовом графа G.

14.11. Пусть $D = (V, T)$ — это глубинный остов, построенный алгоритмом обхода "в глубину" для графа $G = (V, E)$. Докажите, что для каждого обратного ребра $(u, v) \in E \setminus T$ либо u является предком v в D, либо v является предком u в D.

14.12. Модифицируйте алгоритм поиска в глубину так, чтобы он вычислял функцию ВЕРХ(v) и распечатывал список всех мостов графа.

14.13. Обойти (занумеровать) вершины заданного неориентированного графа G с помощью алгоритма обхода "в глубину" и построить дерево этого обхода.
$G = (V, E)$, где $V = \{v_1, v_2, v_3, v_4, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}$,
$E = \{(v_1, v_2), (v_1, v_4), (v_1, v_8), (v_7, v_8), (v_2, v_9), (v_9, v_{11}), (v_3, v_8), (v_6, v_3)$, $(v_3, v_7), (v_6, v_7), (v_{10}, v_9), (v_{10}, v_{11})\}$.

Какое обратное ребро $e \in E \setminus T$ и цикл в G обнаружились в этом обходе первыми? Вычислите для каждой вершины v значение ВЕРХ(v) и определите все мосты графа G.

14.14. Обойти (занумеровать) вершины заданного неориентированного графа G с помощью алгоритма обхода "в глубину" и построить дерево этого обхода.
$G = (V, E)$, где $V = \{a, b, c, d, e, f, g, h, i\}$, а рёбра заданы списками смежности.
$L_a : e, f, h$; $L_b : e, h$; $L_c : f, g, i$; $L_d : c$; $L_e : a, b, h$; $L_f : a, c, g, i$; $L_g : c, f$; $L_h : a, b, e$; $L_i : c, f$

Какое обратное ребро $e \in E \setminus T$ и цикл в G обнаружились в этом обходе первыми? Вычислите для каждой вершины v значение ВЕРХ(v) и определите все мосты графа G.

14.15. Измените алгоритм обхода "в глубину" так, чтобы он позволил перечислить все связные компоненты неориентированного графа.

14.16. Используя алгоритм Дейкстры, определить для заданного нагруженного графа $G = (V, E)$ и выделенной вершины $a \in V$ длины кратчайших путей из этой вершины в остальные вершины G и построить дерево T этих путей.
$V = \{a, b, c, d, e, f\}$, $E = \{(a, b; 154), (a, c; 17), (a, d; 214), (a, e; 63), (b, d; 25), (c, e; 33), (c, d; 192), (c, b; 123), (d, f; 5), (e, f; 140), (d, e; 10)\}$ (здесь каждая скобка $(u, v; D)$ задает ребро $(u, v) \in E$ и его "вес" $c(u, v) = D$). Определите сумму длин всех рёбер дерева T.

14.17. Используя алгоритм Дейкстры, определить для заданного нагруженного графа $G = (V = \{a, b, c, d, e, f\}, E)$ и выделенной вершины $a \in V$ длины кратчайших путей из этой вершины в остальные вершины G и построить дерево T этих путей.

$$C_G = \begin{pmatrix} a & b & c & d & e & f \\ 0 & 25 & 5 & 26 & 53 & 75 \\ 12 & 0 & \infty & \infty & 120 & 40 \\ \infty & 15 & 0 & 20 & 47 & 60 \\ \infty & \infty & \infty & 0 & 30 & 45 \\ \infty & \infty & 75 & 20 & 0 & 20 \\ 40 & 15 & 15 & 26 & \infty & 0 \end{pmatrix}$$

(∞ означает отсутствие ребра.)

Определите сумму длин всех рёбер дерева T.

14.18. Где в доказательстве правильности алгоритма Дейкстры используется неотрицательность весов рёбер? Приведите пример графа (с отрицательными весами), для которого алгоритм Дейкстры дает неверный ответ.

14.19. Докажите, что на каждом шаге алгоритма Дейкстры кратчайший путь из исходной вершины в любую вершину множества S проходит только через вершины множества S.

14.20. Сколько раз может меняться для одной вершины v значение $D[v]$ в ходе работы алгоритма Дейкстры для графа с n вершинами? Привести пример на каждый возможный случай для $n = 6$.

15. Реализация булевых функций с помощью логических схем

Логической схемой (схемой из функциональных элементов) в базисе $B_0 = \{\wedge, \vee, \neg\}$ называется размеченный ориентированный граф без циклов $S = (V, E)$, в котором

1) вершины, в которые не входят рёбра, называются *входами схемы,* и каждая из них помечена некоторой переменной (разным вершинам соответствуют разные переменные);

2) в каждую из остальных вершин входит одно или два ребра; вершины, в которые входит одно ребро помечены функцией $\neg$, а вершины, в которые входят по два ребра, — одной из функций $\wedge$ или $\vee$. Такие вершины называются *функциональными элементами.*

Глубина вершины $v \in V$ в схеме $S = (V, E)$ — это максимальная длина пути из входов S в v.

Глубиной $D(S)$ *схемы* S назовем максимальную из глубин её вершин.

Пусть входы схемы S помечены переменными $x_1, \ldots, x_n$.

С каждой вершиной $v \in V$ схемы S свяжем булеву функцию $f_v(x_1, \ldots, x_n)$, реализуемую в этой вершине. Определим f_v индукцией по глубине v.

Базис. v имеет глубину 0. Тогда это входная вершина, которая помечена некоторой переменной x_i. Положим $f_v(x_1, \ldots, x_n) = x_i$.

Шаг индукции. Пусть всем вершинам w глубины $\leq k$ уже сопоставлены функции f_w и пусть v — произвольная вершина глубины $k+1$. Тогда

а) если v помечена $\neg$ и в неё входит ребро (w, v), то положим $f_v(x_1, \ldots, x_n) = \neg f_w(x_1, \ldots, x_n)$;

б) если v помечена $\wedge$ и в неё входят два ребра (w_1, v) и (w_2, v), то положим

$f_v(x_1,\dots,x_n) = f_{w_1}(x_1,\dots,x_n) \wedge f_{w_2}(x_1,\dots,x_n)$;
в) если v помечена $\vee$ и в неё входят два ребра (w_1, v) и (w_2, v), то положим $f_v(x_1,\dots,x_n) = f_{w_1}(x_1,\dots,x_n) \vee f_{w_2}(x_1,\dots,x_n)$.
Сложность схемы $L(S)$ схемы S — это число функциональных элементов в S.
Сложность $L(f)$ булевой функции $f(x_1,\dots,x_n)$ — это наименьшая из сложностей схем, реализующих эту функцию.
Линейные (неветвящиеся) программы представляют последовательности присваиваний вида
$X = F(X_1,\dots,X_k)$, где $X, X_1,\dots,X_k$ — переменные, F — имя k-местной базисной функции.
Для базиса $B_0 = \{\wedge, \vee, \neg\}$ линейная программа состоит из присваиваний вида $Z = X \wedge Y$, $Z = X \vee Y$ и $Z = \neg X$.
$P_Z(\sigma_1,\dots,\sigma_n)$ — значение, которое вычисляет в переменной Z программа P со входными переменными $X_1,\dots,X_n$ на входе $\sigma_1,\dots,\sigma_n$.
P вычисляет в выходной переменной Z функцию $F(X_1,\dots,X_n)$, если для любого набора значений входов $\sigma_1,\dots,\sigma_n$ после завершения работы P значение $P_Z(\sigma_1,\dots,\sigma_n) = F(\sigma_1,\dots,\sigma_n)$.

Задачи

15.1. Пусть задана логическая схема $S = (V, E)$: $V = \{a(X1), b(X2), c(X3), d(\neg), e(\neg), f(\wedge), g(\wedge), h(\wedge), i(\vee), k(\vee)\}$ (после имени вершины в скобках указана её метка — переменная или булева функция), $E = \{(a, d), (a, g), (b, e), (b, f), (c, f), (c, h), (d, h), (e, g), (f, k), (g, i), (h, i), (i, k)\}$. Какую булеву функцию реализует схема S в вершине k?

15.2. Определите формулу, задающую функцию, которую в вершине v_7 вычисляет следующая логическая схема $S = (V, E)$.
$V = \{x_1, x_2, x_3, x_4, v_1(\vee), v_2(\wedge), v_3(\vee), v_4(\wedge), v_5(\vee), v_6(\neg), v_7(\wedge)\}$ (в скобках указаны функции, приписанные вершинам),
$E = \{(x_1, v_1), (x_1, v_2), (x_2, v_1), (x_3, v_3), (x_3, v_2), (x_4, v_3), (x_4, v_4), (v_1, v_4), (v_4, v_5), (v_2, v_5), (v_3, v_6), (v_5, v_7), (v_6, v_7)\}$.
Постройте линейную программу, вычисляющую ту же функцию.

15.3. Пусть задана логическая схема $S = (V, E)$:
$V = \{a(X), b(Y), c(Z), d(V), e(\wedge), f(\neg), g(\neg), h(\wedge), i(\wedge), k(\neg), m(\vee)\}$
(после имени вершины в скобках указана её метка — переменная или булева функция),
$E = \{(a, e), (b, f), (c, g), (d, e), (d, i), (e, k), (f, h), (g, h), (h, i), (i, m), (k, m)\}$.
Какие из следующих линейных программ вычисляют в переменной Z ту

же функцию $F(X, Y, Z, V)$, что и схема S в вершине m?

$$\begin{array}{lll}
P1: & P2: & P3: \\
V = X \wedge V; & f = \neg Y; & Y = \neg Y; \\
V = \neg V; & g = \neg Z; & Z = \neg Z; \\
Y = \neg Y; & e = X \wedge V; & Z = Y \wedge Z; \\
Z = \neg Z; & k = \neg e; & Z = Z \wedge V; \\
Y = Y \wedge Z; & h = f \wedge g; & V = X \wedge V; \\
Z = Y \wedge V; & i = h \wedge V; & V = \neg V; \\
Z = V \vee Z. & Z = h \vee k. & Z = Z \vee V.
\end{array}$$

15.4. Пусть задана линейная программа P со входными переменными $X1, X2, X3$:

$$\begin{array}{lll}
1) Y = \neg X1; & 4) Y = Y \wedge X2; & 7) Y = W \vee Y; \\
2) Z = \neg X2; & 5) W = X2 \wedge X3; & 8) Z = Z \vee Y. \\
3) U = \neg X3; & 6) Y = Y \wedge U; &
\end{array}$$

Постройте логическую схему S_P со входами $X1, X2, X3$ и функциональными вершинами, соответствующими командам P, вычисляющую ту же функцию, что и P в выходной переменной Z. Чему равна её глубина?

15.5. Докажите, что по каждой линейной программе P со входными переменными $X_1, \ldots, X_n$, вычисляющей в выходной переменной Z некоторую функцию $F(X_1, \ldots, X_n)$, можно эффективно построить логическую схему S_P со входами $X_1, \ldots, X_n$, в которой имеется вершина v такая, что $f_v(X_1, \ldots, X_n) = F(X_1, \ldots, X_n)$.

15.6. Докажите, что минимальная логическая схема для функции сложения по модулю 2 $add(x, y) = (x + y)$ имеет сложность $L(+) = 4$.

15.7. Используя схему SUM_n, постройте логическую схему, реализующую операцию вычитания двух n-разрядных двоичных чисел: $d = a - b$ (при условии, что $a \geq b$). Оцените сложность полученной схемы.

15.8. Определите глубину логических схем S_+, S_{odd}, SUM_1 и SUM_n.

15.9. Два игрока независимо выбирают одно из четырёх чисел от 0 до 3. Первый игрок выигрывает, если выбранные числа совпадают. Постройте логическую схему S_F, определяющую выигрыш 1-го игрока. Её входы x_1, x_2 представляют число, выбранное 1-м игроком, а y_1, y_2 — число, выбранное 2-м игроком. Реализуемая функция $F(x_1, x_2, y_1, y_2)$ равна 1 тогда и только тогда, когда $x_1 = y_1$ и $x_2 = y_2$.

15.10. Постройте логическую схему S_4, определяющую результат голосования в комитете, состоящем из трёх членов и председателя. В случае равенства голосов голос председателя является решающим.

15.11. Постройте логическую схему S_5, определяющую результат голосования в комитете, состоящем из пяти членов (у председателя и секретаря по 2 голоса): $f(x_1, x_2, x_3, x_4, x_5) = 1 \Leftrightarrow \ 2x_1 + 2x_2 + x_3 + x_4 + x_5 \geq 4$. Определите сложность и глубину построенной схемы.

15.12. Постройте логическую схему, определяющую результат умножения двух двухбитовых чисел $a = a_1a_0$ и $b = b_1b_0$. Схема должна иметь 4 выхода, в которых вычисляются биты результата $a * b = c = c_3c_2c_1c_0$. Например, при $a = 10$ и $b = 11$ в результате должно получиться $c_0 = 0, c_1 = 1, c_2 = 1$ и $c_3 = 0$. Определите сложность и глубину построенной схемы.

15.13. Пусть наборы аргументов булевой функции от трёх аргументов упорядочены лексикографически, а её значения задаются последовательностью 8 нулей и единиц. Постройте логические схемы, реализующие следующие функции.
а) $f_1 = (1111\ 1011)$,
б) $f_2 = (1001\ 1001)$,
в) $f_3 = (0011\ 1001)$.

15.14. Постройте логические схемы, реализующие следующие функции:
а) равенство n-битовых чисел:

$$f_n(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n) = 1 \iff x_i = y_i (i = 1, 2, \dots, n);$$

б) отношение “больше” на n-битовых числах:

$$g_n(x_0, x_1, \dots, x_{n-1}, y_0, y_1, \dots, y_{n-1}) = 1 \iff \sum_{i=0}^{n-1} x_i 2^i > \sum_{i=0}^{n-1} y_i 2^i;$$

в) хотя бы две единицы из n

$$t_n(x_1, x_2, \dots, x_n) = 1 \iff \sum_{i=1}^{n} x_i \geq 2.$$

Определите сложность и глубину построенных схем.

15.15. Постройте логические схемы, реализующие следующие функции:
а) $f_n(x_1, x_2, \ldots, x_n) = 1 \iff \sum\limits_{i=1}^{n} x_i \textbf{ mod } 3 = 2$;
б) $f_n(x_1, x_2, \ldots, x_n) = 1 \iff \sum\limits_{i=1}^{n} x_i \textbf{ mod } 4 = 3$;
в) $f_n(x_1, x_2, \ldots, x_n) = 1 \iff \sum\limits_{i=1}^{n} x_i \textbf{ mod } 4 \in \{1, 3\}$.
Определите сложность и глубину построенных схем.

15.16. Пусть ориентированный граф $G = (\{1, 2, 3\}, E)$ задан матрицей смежности $A_G = (a_{i,j}),\ 1 \le i, j \le 3$. Постройте логическую схему, которая по 9 переменным $a_{i,j}$ определяет, является ли G двусвязным, т.е. можно ли в нём из любой вершины достичь две другие вершины. Определите сложность и глубину построенной схемы.

16. Упорядоченные бинарные диаграммы решений (УБДР)

Бинарное дерево решений (БДР) — это бинарное дерево $T = (V, E)$, все внутренние вершины которого помечены переменными, а листья — значениями 0 или 1. Из каждой внутренней вершины v выходят 2 ребра, одно помечено 0, другое — 1; вершина w_0, в которую ведёт ребро, помеченное 0, называется 0-сыном v, а вершина w_1, в которую ведёт ребро, помеченное 1, называется 1-сыном v.
Такое дерево, вершины которого помечены переменными $x_1, \ldots, x_n$, реализует булеву функцию $f(x_1, \ldots, x_n)$, если для каждого набора значений переменных $\sigma_1, \sigma_2, \ldots, \sigma_n$ ветвь в дереве, соответствующая этому набору (из вершины x_i идём по ребру, помеченному σ_i), завершается листом с меткой $f(\sigma_1, \sigma_2, \ldots, \sigma_n)$.

Пусть зафиксирован некоторый порядок n переменных $\pi : x_{\pi(1)}, \ldots, x_{\pi(n)}$.
Упорядоченная бинарная диаграмма решений (УБДР) относительно порядка переменных π — это ориентированный граф без циклов с одним корнем, в котором
1) существуют лишь две вершины, из которых не выходят рёбра; они помечены константами 0 и 1 и называются *стоками*;
2) остальные (внутренние) вершины помечены переменными и из каждой из них выходят два ребра, одно помечено 0, другое — 1;

3) порядок, в котором переменные встречаются на любом пути из корня в сток, совместим с π, т.е. если из вершины, помеченной $x_{\pi(i)}$, есть путь в вершину, помеченную $x_{\pi(j)}$, то $i < j$.
УБДР реализует булеву функцию $f(x_1, \ldots, x_n)$, если для каждого набора значений переменных $\sigma_1, \sigma_2, \ldots, \sigma_n$ путь в диаграмме, начинающийся в корне и соответствующий этому набору (из вершины x_i идём по ребру, помеченному σ_i), завершается стоком с меткой $f(\sigma_1, \sigma_2, \ldots, \sigma_n)$.
Сложностью $L(D)$ *УБДР* D называется *число внутренних вершин* в D.
УБДР называется *сокращённой,* если
1) у любой внутренней вершины v её 0-сын и 1-сын не совпадают;
2) нет такой пары внутренних вершин u и v, для которых поддиаграммы с корнями u и v являются изоморфными (т.е. взаимно однозначно отображаются друг на друга с сохранением всех меток).
Определим два типа эквивалентных преобразований УБДР.
Правило сокращения: если 0-сын и 1-сын вершины v совпадают и равны w, то удалить v, перенаправив все входящие в неё ребра в вершину w.
Правило слияния: если вершины v и w помечены одной переменной и имеют одинаковых 0-сыновей и 1-сыновей, то удалить вершину v, перенаправив все входящие в неё ребра в вершину w.

Теорема. УБДР D является сокращённой тогда и только тогда, когда к ней не применимо ни правило слияния, ни правило сокращения.

Задачи

16.1. Постройте УБДР для функции $odd(X_1, \ldots, X_n) = X_1 + \ldots + X_n$ и оцените её сложность. Сравните со сложностью ДНФ этой функции из задачи 6.13.

16.2. Докажите следующее утверждение о работе алгоритма СОКРАЩЕНИЕ-УБДР:
После выполнения итерации алгоритма для переменной x_i *в полученной диаграмме для каждой подфункции* $f(\sigma_1, \sigma_2, \ldots, \sigma_{i-1}, x_i, \ldots, x_n)$ ($\sigma_k \in \{0, 1\}$ *при* $k = 1, 2, \ldots, i-1$), *существенно зависящей от* x_i, *имеется ровно одна вершина — корень поддиаграммы, реализующая эту подфункцию.*

(Функция $f(x_1, x_2, \ldots, x_i, \ldots, x_n)$ *существенно зависит от переменной* x_i, если существуют такие два набора значений аргументов $(\sigma_1, \ldots, \sigma_{i-1}, 0, \sigma_{i+1}, \ldots, \sigma_n)$ и $(\sigma_1, \ldots, \sigma_{i-1}, 1, \sigma_{i+1}, \ldots, \sigma_n)$, различающиеся только значением x_i, на которых f принимает разные значения:
$f(\sigma_1, \ldots, \sigma_{i-1}, 0, \sigma_{i+1}, \ldots, \sigma_n) \neq f(\sigma_1, \ldots, \sigma_{i-1}, 1, \sigma_{i+1}, \ldots, \sigma_n)$).

16.3. Постройте минимальные УБДР для двуместных функций: $x \wedge y$, $x \vee y$, $x + y$, $x \rightarrow y$, $x|y$.

16.4. Используя алгоритм СОКРАЩЕНИЕ-УБДР, построить сокращённую диаграмму, эквивалентную следующей УБДР $D = (V, E)$:
$V = \{v_1(x_3), v_2(x_1), v_3(x_1), v_4(x_2), v_5(x_2), v_6(x_2), \mathbf{0}, \mathbf{1}\}$ — здесь в скобках после вершин указаны приписанные им переменные;
$E = \{(v_1, v_2; 1), (v_1, v_3; 0), (v_2, v_4; 1), (v_2, v_5; 0), (v_3, v_4; 0), (v_3, v_6; 1), (v_4, \mathbf{0}; 1), (v_4, \mathbf{1}; 0), (v_6, \mathbf{0}; 0), (v_6, \mathbf{1}; 1), (v_5, \mathbf{0}; 1), (v_5, \mathbf{1}; 0)\}$ (третий параметр после ; — метка ребра).
Какую функцию реализует полученная схема? Постройте её таблицу.

16.5. Постройте минимальные УБДР для функции
$f(x_1, x_2, x_3, x_4, x_5, x_6) = (x_1 \wedge x_2) + (x_3 \wedge x_4) + (x_5 \wedge x_6)$
относительно двух упорядочений переменных:
а) $x_1 < x_2 < x_3 < x_4 < x_5 < x_6$;
б) $x_1 < x_3 < x_5 < x_2 < x_4 < x_6$.

16.6. Пороговая функция T_n^k от n переменных с порогом k выдаёт 1, если во входном наборе имеется не менее k единиц: $T_n^k(x_1, x_2, \ldots, x_n) = 1 \Leftrightarrow x_1 + x_2 + \ldots + x_n \geq k$.
а) Постройте минимальные УБДР для пороговых функций T_3^2, T_4^2, T_5^3.
б) Зависит ли сложность минимальной УБДР для пороговых функций от порядка переменных?
в) Оцените сложность минимальной УБДР для пороговой функции T_n^k.

16.7. Докажите, что сложность сокращённой УБДР для функции
$$p_n(x_1, x_2, \ldots, x_{2n-1}, x_{2n}) = (x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee \ldots \vee (x_{2n-1} \wedge x_{2n})$$
равна
а) $2n + 2$ при упорядочении переменных $\pi_1 : x_1, x_2, \ldots x_{2n-1}, x_{2n}$;
б) 2^{n+1} при упорядочении переменных $\pi_2 : x_1, x_{n+1}, x_2, x_{n+2} \ldots x_n, x_{2n}$.

16.8. Выберите подходящий порядок переменных и постройте для него минимальные УБДР, реализующие функции из задач 15.9, 15.10 и 15.11.

16.9. Как мы видели, логические схемы естественным образом реализуются в виде неветвящихся программ. Наоборот, для деревьев решений и УБДР естественным программным представлением являются *ветвящиеся программы*, включающие лишь условные операторы вида **if ... then ... else ...** с тестами вида “$x = 0$?” и “$x = 1$?” (они соответствуют внутренним вершинам диаграмм) и операторы присваивания значения 0 или 1 результату **return(0)** и **return(1)** (они соответствуют вершинам-стокам).

Напишите ветвящиеся программы, вычисляющие функции, представляемые УБДР D_1 на рис. 7 и D_2 и на рис. 8.

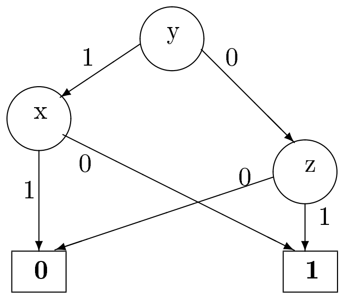


Рис. 7. УБДР D_1 для функции $f_1(x, y, z)$

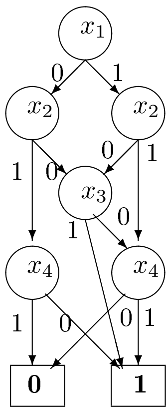


Рис. 8. УБДР D_2 для функции $f_2(x_1, x_2, x_3, x_4)$

16.10. Выберите подходящий порядок переменных и постройте УБДР для функций f_n, g_n и t_n из пунктов (а), (б) и (в) задачи 15.14 и оцените их сложность.

16.11. Постройте УБДР для функций f_n, g_n и t_n из пунктов (а), (б) и (в) задачи 15.15 и оцените их сложность.

16.12. Пусть , как и в задаче 15.16, ориентированный граф $G = (\{1, 2, 3\}, E)$ задан матрицей смежности $A_G = (a_{i,j}),\ 1 \le i, j \le 3$. Выберите подходящий порядок переменных $a_{i,j}$ и постройте УБДР, которая по их значениям определяет является ли G двусвязным, т.е. можно ли в нём из любой вершины достичь две другие вершины. Определите сложность построенной диаграммы.

16.13. а) Пусть D — сокращённая УБДР для функции $f(x_1, \ldots, x_n)$ при некотором порядке переменных π, $1 \le i \le n$, и $\sigma \in \{0, 1\}$. Предложите процедуру, которая перестроит D в УБДР D' для функции $f'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = f(x_1, \ldots, x_{i-1}, \sigma, x_{i+1}, \ldots, x_n)$ при том же порядке π, из которого удалена x_i. Всегда ли полученная УБДР D' будет сокращённой?
б) Примените процедуру из пункта (а) для получения из УБДР D_2 на рис. 8 диаграмм для следующих функций: $f_2(0, x_2, x_3, x_4), f_2(x_1, 1, x_3, x_4), f_2(x_1, x_2, 0, x_4)$.

17. Конечные автоматы: преобразователи и распознаватели

Конечный автомат–преобразователь — это система вида

$$A = \langle \Sigma_X, \Sigma_Y, Q, q_0, \Phi, \Psi \rangle,$$

включающая следующие компоненты:
$\Sigma_X = \{a_1, \ldots, a_m\}$ $(m \ge 1)$ — *входной алфавит*;
$\Sigma_Y = \{b_1, \ldots, b_r\}$ $(r \ge 1)$ — *выходной алфавит*;
$Q = \{q_0, \ldots, q_{n-1}\}$ $(n \ge 1)$ — *алфавит внутренних состояний*;
$q_0 \in Q$ — *начальное состояние* автомата;
$\Phi : Q \times \Sigma_X \to Q$ — *функция переходов*: $\Phi(q, a)$ — это состояние, в которое переходит автомат из состояния q, когда получает на вход символ a;
$\Psi : Q \times \Sigma_X \to \Sigma_Y$ — *функция выходов*, $\Psi(q, a)$ — это символ из Σ_Y, который выдаёт на выход автомат в состоянии q, когда получает на вход символ a.

Диаграмма автомата-преобразователя $A = \langle \Sigma_X, \Sigma_Y, Q, q_0, \Phi, \Psi \rangle$ — это мультиграф $D_A = (Q, E)$ с помеченными рёбрами, в котором выделена вершина-начальное состояние q_0 и из каждой вершины $q \in Q$ выходит $|\Sigma_X|$ рёбер, помеченных парами символов a/b $(a \in \Sigma_X, b \in \Sigma_Y)$. Таким образом, для каждой вершины $q \in Q$ и каждого символа $a \in \Sigma_X$ имеется единственное ребро с меткой $a/\Psi(q, a)$ из q в вершину $q' = \Phi(q, a)$.

Работа автомата-преобразователя характеризуется последовательностью проходимых им состояний $q(0), q(1), \ldots, q(t), \ldots$ и последовательностью выходных символов $y(1), \ldots, y(t), \ldots$.
Они определяются следующими реккурентными соотношениями:
$q(0) = q_0$
$q(1) = \Phi(q(0), x_1)$
$y(1) = \Psi(q_0, x_1)$
$\ldots$
$q(t+1) = \Phi(q(t), x_t)$
$y(t+1) = \Psi(q(t), x_t)$
$\ldots$
Слово в алфавите $\Sigma = \{a_1, \ldots, a_m\}$ — это конечная последовательность символов этого алфавита: $w = w_1 \ldots w_n$, $w_i \in \Sigma$ при $i = 1, \ldots, n$. Число букв в этой последовательности называется *длиной слова* и обозначается $|w|$.
ε — "пустое" слово длины 0.
Σ^* — множество всех слов в алфавите Σ (в том числе и пустое слово ε).
Язык в алфавите Σ — произвольное подмножество Σ^*.

Детерминированный конечный автомат-распознаватель (ДКА) — это система вида

$$A = \langle \Sigma, Q, q_0, F, \Phi \rangle,$$

включающая следующие компоненты:
$\Sigma = \{a_1, \ldots, a_m\}$ $(m \geq 1)$ — конечное множество - *входной алфавит*;
$Q = \{q_0, \ldots, q_{n-1}\}$ $(n \geq 1)$ — конечное множество - *алфавит внутренних состояний*;
$q_0 \in Q$ — *начальное состояние* автомата;
$F \subseteq Q$ — множество *принимающих (допускающих, заключительных)* состояний;
$\Phi : Q \times \Sigma \to Q$ — *функция переходов*, $\Phi(q, a)$ — это состояние, в которое переходит автомат из состояния q, когда получает на вход символ a.
Функцию Φ называют *программой* автомата A и задают как список из $n \cdot m$ *команд* вида $q_i a_j \ \to \ \Phi(q_i, a_j)$ $(q_i \in Q, a_j \in \Sigma)$.
Её можно задавать также с помощью таблицы размера $n \times m$, строки которой соответствуют состояниям из Q, а столбцы — символам из входного алфавита Σ, и в которой на пересечении строки q_i и столбца a_j стоит состояние $\Phi(q_i, a_j)$.

Диаграмма ДКА $A = \langle \Sigma, Q, q_0, F, \Phi \rangle$ — это мультиграф $D_A = (Q, E)$ с помеченными рёбрами, в котором выделена вершина-начальное состояние q_0 и из каждой вершины $q \in Q$ выходит $|\Sigma|$ рёбер, помеченных символами $a \in \Sigma$ так, что для каждой вершины $q \in Q$ и каждого символа $a \in \Sigma$ имеется единственное ребро из q в вершину $q' = \Phi(q, a)$ с меткой a.

Конфигурация ДКА $A = \langle \Sigma, Q, q_0, F, \Phi \rangle$ — пара вида (q, w), в которой $q \in Q$ и $w \in \Sigma^*$. Отношение $\vdash_A$ перехода за 1 шаг
$(q, w) \vdash_A (q', w')$ тогда и только тогда, когда $w = aw'$, $\Phi(q, a) = q'$.
$(q, \varepsilon) \vdash_A (q, \varepsilon)$.
$\vdash_A^*$ — обозначим рефлексивное и транзитивное замыкание $\vdash_A$.
ДКА A *распознаёт (допускает, принимает) слово* w, если $(q_0, w) \vdash_A^* (q, \varepsilon)$ для некоторого $q \in F$, т.е. после обработки слова w автомат переходит в принимающее состояние.
Язык L_A*, распознаваемый (допускаемый, принимаемый)* автоматом A, состоит из всех слов, распознаваемых этим автоматом: $L_A = \{w \mid A$ распознает $w\}$. Язык называется *конечно автоматным* или просто *автоматным*, если он распознаётся некоторым ДКА.
Автоматы A и B называются *эквивалентными*, если совпадают распознаваемые ими языки, т.е. $L_A = L_B$.
В терминах диаграмм:
A *распознаёт (допускает, принимает) слово* w, если в диаграмме D_A имеется путь из q_0 в $q \in F$, который несёт слово w, т.е. w переводит q_0 в распознающее состояние q. Язык L_A, распознаваемый автоматом A, состоит из всех слов, которые переводят в его диаграмме D_A начальное состояние q_0 в заключительные состояния из F.
Для $q \in Q$ через $L(q)$ обозначим множество слов, которые переводят q_0 в q. Тогда $L_A = \bigcup_{q \in F} L(q)$.

Пусть ДКА $M_1 = \langle \Sigma, Q_1, q_0^1, F_1, \Phi_1 \rangle$ и $M_2 = \langle \Sigma, Q_2, q_0^2, F_2, \Phi_2 \rangle$ распознают языки L_1 и L_2, соответственно. Определим по ним автомат $M = M_1 \times M_2$, называемый *произведением* M_1 и M_2: $M = \langle \Sigma, Q, q_0, F, \Phi \rangle$, где
$Q = Q_1 \times Q_2 = \{(q, p) \mid q \in Q_1,\ p \in Q_2\}$,
$q_0 = (q_0^1, q_0^2)$,
$\Phi((q, p), a) = (\Phi_1(q, a), \Phi_2(p, a))$, для всех $(q, p) \in Q$ и $a \in \Sigma$,
множество принимающих состояний F определяется в зависимости от операции над языками L_1 и L_2, которую должен реализовать M. Именно,
а) при $F = F_\cup = \{(q, p) \mid q \in F_1 \text{ или } p \in F_2\}$ $L_M = L_1 \cup L_2$;
б) при $F = F_\cap = \{(q, p) \mid q \in F_1 \text{ и } p \in F_2\}$ $L_M = L_1 \cap L_2$;
в) при $F = F_\setminus = \{(q, p) \mid q \in F_1 \text{ и } p \notin F_2\}$ $L_M = L_1 \setminus L_2$.

Недетерминированный конечный автомат (НКА) — это система вида
$$M = \langle \Sigma, Q, q_0, F, \Phi \rangle,$$
где компоненты: Σ, Q, q_0, F те же, что и у ДКА, а функция переходов $\Phi : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$ для каждого $q \in Q$ и $a \in \Sigma$ задаёт подмножество состояний (возможно пустое!) $\Phi(q, a) \subseteq Q$, в каждое из которых может перейти автомат из состояния q, когда читает символ a. $\Phi(q, \varepsilon)$ — это множество состояний, в каждое из которых может перейти автомат из состояния q без чтения символа на входе.

В диаграмме $D_M = (Q, E)$ НКА $M = \langle \Sigma, Q, q_0, F, \Phi \rangle$ для каждой вершины $q \in Q$ и $a \in \Sigma$ рёбра с меткой a идут во все вершины множества $\Phi(q, a)$, а рёбра с меткой ε идут во все вершины множества $\Phi(q, \varepsilon)$.
Отношение $\vdash_M$ *перехода из одной конфигурации в другую за один шаг для НКА*:
$(q, w) \vdash_M (q', w')$ тогда и только тогда, когда $(w = aw'$ *и* $q' \in \Phi(q, a))$ или $(w = w'$ и $q' \in \Phi(q, \varepsilon))$.
Как и для ДКА, через $\vdash_M^*$ обозначим рефлексивное и транзитивное замыкание отношения $\vdash_M$.
Определения распознавания слов и языков для НКА совпадает с соответствующими определениями для ДКА; $w \in L_M$ тогда и только тогда, когда $(q_0, w) \vdash_M^* (q, \varepsilon)$ для некоторого $q \in F$.
Отличие состоит в том, что у НКА может быть несколько различных путей вычисления на одном и том же входном слове w. НКА распознает (допускает, принимает) это слово, если *хотя бы один* из этих способов приводит в принимающее состояние из F.
В терминах диаграмм:
НКА M распознает слово w тогда и только тогда, когда в D_M имеется путь из q_0 в некоторое $q \in F$, на рёбрах которого написано слово w (с точностью до меток ε).

Задачи

17.1. Пусть задан конечный автомат-преобразователь $\mathbf{A} = \langle \Sigma_X = \{0, 1\}, \Sigma_Y = \{\text{A}, \text{P}, \text{T}\}, Q = \{0, 1, 2, 3\}, 0, \Phi, \Psi \rangle$, где Φ/Ψ:

$Q \backslash \Sigma_X$	0	1
0	0 / T	1 / A
1	2 / P	3 / T
2	1 / A	2 / T
3	1 / A	3 / P

а) В какое выходное слово автомат **A** преобразует слово 010100?
б) Какое входное слово автомат **A** перерабатывает в выходное слово APAPAT?

17.2. Следующий конечный автомат-преобразователь $MINUS_c = \langle \Sigma_X = \{0, 1\}, \Sigma_Y = \{0, 1\}, Q = \{0, 1, 2, 3\}, 0, \Phi, \Psi \rangle$, где Φ/Ψ:

$Q \backslash \Sigma_X$	0	1
0	3/ 1	1 /0
1	1 / 1	2 / 0
2	2 / 0	2 / 1
3	1 / 0	1 / 1

вычитает из входного двоичного числа x некоторую константу c и выдает при $c \leq x$ выходное двоичное число $y = x - c$. Чему равна эта константа c?

17.3. Автомат по продаже кофе имеет щель для получения монет, кнопку, нажатие которой после уплаты достаточной суммы приводит к получению кофе, и накопитель, через который он выдаёт сдачу покупателю. Автомат принимает монеты достоинством в 1, 2 и 5 рублей. Чашка кофе стоит 8 руб. Пока полученная сумма недостаточна, горит красная лампочка. Если сумма, полученная автоматом, ≥ 8, то зажигается зелёная лампочка и после нажатия кнопки автомат наливает кофе и, если требуется, даёт сдачу. Если автомат получает монету, когда горит зелёная лампочка, то он немедленно её возвращает. Определите входной и выходной алфавиты конечного автомата, управляющего продажей кофе, и постройте его функции переходов и выходов.

17.4. Электронные часы имеют табло с указанием часов, минут и секунд и две управляющие кнопки. Одна кнопка переводит часы из нормального режима в режим настройки времени — вначале в настройку часов, затем — минут, затем — секунд, а затем возвращает в нормальный режим. Другая кнопка в нормальном режиме ничего не меняет, а в режиме настройки нажатие на неё увеличивает на единицу число настраеваемых часов, минут или секунд. Постройте автомат, который принимает на вход сигналы нажатия от двух кнопок, а на выходе выдаёт сигналы изменения режима и увеличения соответствующего числа.

17.5. Пусть $\Sigma_X = \{a, b\}$, $Q = \{0, 1, 2, 3, 4, 5\}$, $F = \{3, 4\}$ и функция переходов конечного автомата-распознавателя $\mathbf{A} = \langle \Sigma_X, Q, 0, F, \Phi \rangle$ задана таблицей

Φ:

$Q \backslash \Sigma_X$	a	b
0	1	2
1	1	3
2	4	2
3	1	5
4	5	2
5	5	5

Какие из следующих трёх слов распознаются автоматом $\mathbf{A}$?
$W = aaabbabab, V = babbbabba, U = ababaaab$.

17.6. Какие из следующих трёх конечных автоматов $A_i = \langle\{a,b\},\{0,1,2,3,4\},0,F = \{1\},\Phi_i\rangle$ $(i = 1,2,3)$ распознают язык L, состоящий из всех слов, которые начинаются на b и содержат число букв a, кратное 3?

Φ_1				Φ_2				Φ_3		
$Q\backslash\Sigma$	a	b		$Q\backslash\ \Sigma$	a	b		$Q\backslash\Sigma$	a	b
0	4	1		0	3	1		0	4	1
1	2	2		1	1	2		1	2	1
2	3	1		2	4	2		2	3	2
3	1	3		3	3	3		3	1	3
4	4	4		4	1	4		4	4	4

17.7. Постройте детерминированные конечные автоматы, которые распознают следующие языки в алфавите $\Sigma = \{a,b\}$:
а) $L_1 = \{w \mid$ длина w делится на $5\}$;
б) $L_2 = \{w \mid w$ не содержит подслов aab и $bba\}$;
в) $L_3 = \{w \mid w$ содержит чётное число букв a и нечётное число букв $b\}$;
г) $L_4 = \{w \mid$ число букв a делится на 3, а число букв b — на $2\}$.

17.8. Постройте автомат-распознаватель, который проверяет правильность сложения. На вход поступают последовательности троек нулей и единиц:
$(x_1(1),x_2(1),y(1)),(x_1(2),x_2(2),y(2)),\ldots,(x_1(n),x_2(n),y(n))$.
Автомат должен допустить такую последовательность, если $y = y(n)\ldots y(2)y(1)$ — это первые n битов суммы двоичных чисел $x_1 = x_1(n)\ldots x_1(2)x_1(1)$ и $x_2 = x_2(n)\ldots x_2(2)x_2(1)$.

17.9. Докажите лемму о произведении конечных автоматов:
Для любых двух состояний (q,p) и (q',p') автомата $M = M_1 \times M_2$ и любого входного слова w слово w переводит (q,p) в (q',p') в автомате M тогда и только тогда, когда оно переводит q в q' в автомате M_1 и p в p' в автомате M_2.

17.10. Ниже приведён конечный автомат-распознаватель $\mathbf{A} = \langle\Sigma_X = \{a,b\}, Q = \{0,1,2,3\},\ 0, F = \{3\},\Phi\rangle$, у которого Φ:

$Q\backslash\Sigma_X$	a	b
0	1	0
1	1	2
2	3	0
3	1	2

Докажите, что этот автомат распознаёт язык, состоящий из всех слов в алфавите $\{a, b\}$, заканчивающихся на aba.

17.11. Пусть $L_1 = \{w \mid \text{число букв } b \text{ в } w \text{ кратно } 3\}$ и $L_2 = \{w \mid w$ не содержит подслово $aaa\}$ — два языка в алфавите $\Sigma = \{a, b\}$.
а) Постройте конечные автоматы **A** и **B**, распознающие языки L_1 и L_2, соответственно.
б) Постройте произведение этих автоматов $\mathbf{C} = \mathbf{A} \times \mathbf{B}$ и задайте множества его допускающих состояний F_1, F_2 и F_3, для распознавания языков $R_1 = L_1 \cup L_2$, $R_2 = L_1 \cap L_2$, $R_3 = L_1 \setminus L_2$, соответственно.

17.12. Пусть задан недетерминированный конечный автомат (без пустых переходов) $M = \langle \{ a, b \}, \{ 0, 1, 2, 3 \}, 0, \{ 1, 3 \}, \Phi \rangle$ с программой Φ: $0a \to 0, 0a \to 1, 0b \to 2$, $1b \to 0$, $2a \to 0, 2a \to 3, 2b \to 1, 2b \to 2, 2b \to 3$, $3a \to 0, 3a \to 3$. а) Определите, какие из следующих слов распознаются автоматом M: $w_1 = aaabab$, $w_2 = ababbba$, $w_3 = bbaabba$, $w_4 = bbbaab$.
б) Постройте детерминированный конечнеый автомат, эквивалентный M.

17.13. Ниже приведён недетерминированный конечный автомат
$M = \langle \Sigma = \{0, 1\}, Q = \{0, 1, 2, 3, 4, 5\}, 0, F = \{5\}, \Phi \rangle$, где Φ:

$Q \backslash \Sigma$	0	1	ε
0	$\varnothing$	1, 5	$\varnothing$
1	$\varnothing$	$\varnothing$	2, 3
2	5	4	$\varnothing$
3	3	$\varnothing$	4
4	4, 5	$\varnothing$	5
5	$\varnothing$	$\varnothing$	$\varnothing$

а) Определите, какие из следующих слов распознаются автоматом M: $W = 101, V = 110000, U = 01, X = 10, Y = 11, R = 1$.
б) Докажите, что $L_M = \{10^i \mid i \geq 0\} \cup \{110^i \mid i \geq 0\}$.
в) Постройте детерминированный конечнеый автомат, эквивалентный M.

17.14. Используя процедуру детерминизации недетерминированных автоматов, постройте ДКА, эквивалентные следующим НКА M_1 и M_2.
а) $M_1 = \langle \{a, b\}, \{0, 1, 2\}, 0, \{2\}, \Phi \rangle$ с программой $\Phi : 0a \to 1, 0\varepsilon \to 1$, $1b \to 2$, $1b \to 1$, $1\varepsilon \to \ 2, 2b \to \ 2$.
б) $M_2 = \langle \{a, b\}, \{0, 1, 2\}, 0, \{2\}, \Phi \rangle$ с программой $\Phi :\ \ 0a \to 1,\ 0a \to 2$, $1b \to 2,\ 1\varepsilon \to 2,\ 2b \to\ 0$.

17.15. Пусть задан недетерминированный конечный автомат (без пустых переходов) $M = \langle\{0,1\},\{q,p,s\},q,F=\{p\},\Phi\rangle$ с программой
$\Phi : q0 \to p, q0 \to s, q1 \to q, p0 \to q, p0 \to p, s1 \to q, s1 \to p.$

Какие из следующих трёх ДКА эквивалентны M?

$M1 = \langle\{0,1\},\{q,ps,pq,pqs\},q,F1=\{ps,pq,pqs\},\Phi_1\rangle$ с программой Φ_1: $q0 \to ps, q1 \to q, ps0 \to pq, ps1 \to pq, pq0 \to pqs, pq1 \to q, pqs0 \to pqs, pqs1 \to pq.$

$M2 = \langle\{0,1\},\{q,p,s,ps,qs,pq,qps,\varnothing\},q,F2=\{p,ps,pq,pqs\},\Phi_2\rangle$ с программой Φ_2:
$q0 \to ps, q1 \to q, p0 \to pq, p1 \to q, ps0 \to qs, ps1 \to pq, pq0 \to pqs, pq1 \to q,$ $qs0 \to ps, qs1 \to q, pqs0 \to pqs, pqs1 \to pq, \varnothing 0 \to \varnothing, \varnothing 1 \to \varnothing.$

$M3 = \langle\{0,1\},\{q,p,s,ps,qs,pq,qps,\varnothing\},q,F3=\{p,ps,pq,pqs\},\Phi_3\rangle$ с программой Φ_3:
$q0 \to ps, q1 \to q, p0 \to pq, p1 \to q, s0 \to \varnothing, s1 \to pq, ps0 \to pq, ps1 \to pq, pq0 \to pqs,\ pq1 \to q, qs0 \to ps, qs1 \to q, pqs0 \to pqs, pqs1 \to pq, \varnothing 0 \to \varnothing, \varnothing 1 \to \varnothing.$

17.16. Постройте детерминированные конечные автоматы, распознающие следующие языки в алфавите $\{0,1\}$.
L_1 – все слова, в которых на 4-ом с конца месте стоит 1.
L_2 – все слова, содержащие подслова 111 и 010 и не содержащие подслово 001.
L_3 – все слова, заканчивающиеся на 10, 11 или 01.
Указание: вначале постройте соответствующие НКА, а затем их детерминизируйте.

18. Регулярные языки и конечные автоматы

Конкатенация языков.
Пусть L_1 и L_2 — языки в алфавите Σ.
Тогда $L = L_1 \circ L_2 = \{w \mid (\exists w_1 \in L_1)(\exists w_2 \in L_2)(w = w_1 w_2)\}$.
"Степени" языка L: $L^0 = \{\varepsilon\}$, $L^1 = L, \ldots,\ L^{i+1} = L \circ L^i \quad (i = 1, 2, \ldots)$.
Таким образом в L^i входят все слова, которые можно разбить на i подряд идущих слов из L.
Итерация L^* языка L — все слова которые можно разбить на несколько подряд идущих слов из L:
$(L)^* = \{\varepsilon\} \cup \{w \mid (\exists k \geq 1)(w = w_1 w_2 \ldots w_k)$ *и все* $w_i \in L\}$.

Ее можно представить с помощью степеней: $L^* = \bigcup_{i=0}^{\infty} L^i$.
$L^+ = \bigcup_{i=1}^{\infty} L^i$ — удобное сокращение для выражения $L \circ L^*$.

Регулярные выражения над алфавитом Σ и представляемые ими языки.

Выражение r	Язык L_r
$\varnothing$	$L_\varnothing = \varnothing$
ε	$L_\varepsilon = \{\varepsilon\}$
$a \in \Sigma$	$L_a = \{a\}$
Пусть r_1 и r_2 — это регулярные выражения. Тогда следующие выражения являются регулярными	L_{r_1} и L_{r_2} — представляемые ими языки. и представляют языки:
$r = (r_1 + r_2)$	$L_r = L_{r_1} \cup L_{r_2}$
$r = (r_1 \circ r_2)$	$L_r = L_{r_1} \circ L_{r_2}$
$r = r_1^*$	$L_r = L_{r_1}^*$

Два регулярных выражения r и p называются *эквивалентными*, если совпадают представляемые ими языки, т.е. $L_r = L_p$. В этом случае пишем $r = p$.

Задачи

18.1. Определите конкатенацию для следующих пар языков L_1 и L_2:
а) $L_1 = \{a, ab, abb\}$ и $L_2 = \{\varepsilon, a, b, ab, ba\}$;
б) $L_1 = \{\varepsilon, a, ab, abb\}$ и $L_2 = \{a, b, abb, ba\}$;
в) $L_1 = \{\varepsilon, a, b, ab, aba\}$ и $L_2 = \{\varepsilon, a, b, ab, ba\}$.

18.2. Пусть $L = \{baa, bab, bba, bbb\}$. Какой из следующих языков является итерацией L^* этого языка?
а) $\{w \mid w = bw' \text{ и } |w| \text{ делится на 3 } \} \cup \{\varepsilon\}$;
б) $\{w \mid w = bw' \text{ и } |w| \geq 3\} \cup \{\varepsilon\}$;
в) $\{w \mid w = w_1w_2w_3 \ldots w_{3n} \text{ и } w_{3i+1} = b \text{ для всех } i < n\} \cup \{\varepsilon\}$;
г) $\{w \mid w = bw' \text{ и } |w| \geq 12\}$.

18.3. Какие из следующих регулярных выражений задают все слова из нулей и единиц, в которых нет двух подряд идущих 0?
а) $(1 + 01)^*(\varepsilon + 0)$;
б) $(1^*01^*)^*$;
в) $(01)^*1^*01^*$;
г) $1^*01(1 + 01)^*(\varepsilon + 0)$;
д) $(1 + 01)^*(0 + 1)$;
е) $1^*(011^*)^*(\varepsilon + 0)$.

18.4. Пусть регулярное выражение $b(ab)^*$ определяет некоторый язык над алфавитом $S = \{a, b\}$. Какие из следующих регулярных выражений задают тот же язык?
а) $a(ba)^*$;
б) $(ba)^*b$;
в) b^*ab^*;
г) $b(ab + abab)^*$;
д) $(ba)^*b(ab)^*$;
е) $(ab + ba)^*$.

18.5. Докажите, что регулярное выражение $(1 + 01 + 001)^*(\varepsilon + 0 + 00)$ представляет язык, состоящий из всех слов в алфавите $\{0, 1\}$, которые не содержат подслово 000.

18.6. Докажите следующие эквивалентности для регулярных выражений.
а) $p^*(p + q)^* = (p + qp^*)^* = (p + q)^*$;
б) $p(qp)^* = (pq)^*p$;
в) $(p^*q^*)^* = (q^*p^*)^*$;
г) $(pq)^+(q^*p^* + q^*) = (pq)^*pq^+p^*$.

18.7. Постройте регулярное выражение, задающее язык L в алфавите $\Sigma = \{0, 1\}$.
а) $L_1 = \{w \mid w$ содержит нечётное число букв 0 и чётное число букв $1\}$;
б) $L_2 = \{w \mid w$ содержит подслово 001 или подслово $110\}$;
в) $L_3 = \{w \mid w$ содержит по крайней мере мере два подряд идущих $0\}$;
г) $L_4 = \{w \mid w$ не содержит подслов 011 и $010\}$.

18.8. Определите, какой язык представляется следующими регулярными выражениями.
а) 0^*1^*0;
б) $(01^*)^*0$;
в) $(00 + 11 + (01 + 10)(00 + 11)^+(01 + 10))^*$.

18.9. Упростить следующие регулярные выражения.
а) $(00^*)0 + (00)^*$;
б) $(0 + 1)(\varepsilon + 00)^+ + (0 + 1)$;
в) $(0 + \varepsilon)0^*1$;
г) $(0^* + 0)^*(01^* + 0)$.

18.10. Выше в задаче 17.8 предлагалось построить автомат-распознаватель, который проверяет правильность сложения. Постройте регулярное выражение, задающее распознаваемый этим автоматом язык L, т.е. следующее множество слов в алфавите $\{0,1\}^3$:
$L = \{(x_1(1), x_2(1), y(1))(x_1(2), x_2(2), y(2)) \dots (x_1(n), x_2(n), y(n)) \mid$ $y = y(n) \dots y(2)y(1)$ — это первые n битов суммы двоичных чисел $x_1 = x_1(n) \dots x_1(2)x_1(1)$ и $x_2 = x_2(n) \dots x_2(2)x_2(1)\}$.

18.11. Пусть M_r — это автомат, который строится в доказательстве теоремы 15.1 по регулярному выражению r. Докажите, что
а) у M_r нет переходов из единственного заключительного состояния q_f;
б) в диаграмме M_r из каждой вершины выходит не более двух рёбер;
в) число состояний M_r не более чем вдвое превосходит длину выражения r, т.е. $|Q| \le 2|r|$.

18.12. Какие из следующих трёх автоматов $C1, C2, C3$ распознают язык, представляемый регулярным выражением $(00+1)^*1$?
$C1 = \langle \{0,1\}, \{q,p,r,s,t\}, q, F1 = \{t\}, \Phi_1 \rangle$,
$C2 = \langle \{0,1\}, \{q,p,r,s\}, q, F2 = \{s\}, \Phi_2 \rangle$,
$C3 = \langle \{0,1\}, \{q,p,r,s,t\}, q, F3 = \{s\}, \Phi_3 \rangle$.
Программы автоматов заданы в следующих таблицах ($\varnothing$ означает отсутствие соответствующего перехода).

Φ_1					Φ_2				Φ_3		
Q/Σ	0	1	ε		Q/Σ	0	1		Q/Σ	0	1
q	p	r	s		q	p	r		q	p	s
p	r	$\varnothing$	$\varnothing$		p	r	$\varnothing$		p	r	t
r	$\varnothing$	$\varnothing$	q,s		r	p	r,s		r	p	s
s	$\varnothing$	t	$\varnothing$		s	$\varnothing$	$\varnothing$		s	p	s
t	$\varnothing$	$\varnothing$	$\varnothing$						t	t	t

18.13. Какие из следующих трёх автоматов $C1, C2, C3$ распознают язык, представляемый регулярным выражением $0(10+1)^*$?
$C1 = \langle \{0,1\}, \{q,p,r,s,t\}, q, F_1 = \{t\}, \Phi_1 \rangle$,
$C2 = \langle \{0,1\}, \{q,p,r,s\}, q, F_2 = \{p,r\}, \Phi_2 \rangle$,
$C3 = \langle \{0,1\}, \{q,p,r,s,t\}, q, F_3 = \{p,r,s\}, \Phi_3 \rangle$.
Программы автоматов заданы в следующих таблицах ($\varnothing$ означает отсутствие соответствующего перехода).

Φ_1					Φ_2				Φ_3		
Q/Σ	0	1	ε		Q/Σ	0	1		Q/Σ	0	1
q	p	$\varnothing$	$\varnothing$		q	p	$\varnothing$		q	p	t
p	$\varnothing$	$r,\ s$	$\varnothing$		p	$\varnothing$	$r,\ s$		p	t	s
r	$\varnothing$	$\varnothing$	$p,\ t$		r	$\varnothing$	r,s		r	t	s
s	r	$\varnothing$	$\varnothing$		s	r	$\varnothing$		s	r	s
t	$\varnothing$	$\varnothing$	$\varnothing$						t	t	t

18.14. Постройте детерминированные конечные автоматы, распознающие языки, задаваемые следующими регулярными выражениями.
а) $r_1 = aa(bb + aa)^*bb$,
б) $r_2 = (a + bb)(a + b)^*(aa + bb)$,
в) $r_3 = (a^+ + b^+)(ab + ba)^*$,
г) $r_4 = (abb)^*(aab^* + bba^*)$,
д) $r_5 = b^*(ab^*a)^*b^*$.

19. Свойства замкнутости класса автоматных языков. Неавтоматные языки.

Пусть Σ и Δ — два алфавита. Отображение $\phi :\ \Sigma^* \to \Delta^*$ слов первого из них в слова второго называется *гомоморфизмом,* если для любых двух слов w_1 и w_2 в алфавите Σ имеет место равенство $\phi(w_1w_2) = \phi(w_1)\phi(w_2)$.
Из этого определения следует, что $\phi(\varepsilon) = \varepsilon$ и для каждого слова $w = a_1a_2\dots a_n$ $(a_i \in \Sigma, i = 1, 2, \dots n)$ $\phi(w) = \phi(a_1)\phi(a_2)\dots\phi(a_n)$.
Пусть $\phi :\ \Sigma^* \to \Delta^*$ — произвольный гомоморфизм и L — язык в алфавите Σ. *Образом* $\phi(L)$ языка L при гомоморфизме ϕ называется язык $\phi(L) = \{\phi(w) \mid w \in L\}$, состоящий из образов всех слов языка L.
Пусть L — язык в алфавите Δ. *Прообразом* этого языка при гомоморфизме ϕ называется язык $\phi^{-1}(L) = \{w \in \Sigma^* \mid \phi(w) \in L\}$, состоящий из всех таких слов в алфавите Σ, чьи образы при гомоморфизме ϕ попадают в L.

Пусть $\Delta \subset \Sigma$. Проекцией языка L в алфавите Σ на подалфавит Δ называется язык
$PROJ_\Delta(L) = \{w \mid w$ получено из некоторого слова $v \in L$ вычёркиванием всех символов, не принадлежащих алфавиту $\Delta\}$.

Класс автоматных языков замкнут относительно теоретико-множественных операций объединения, пересечения, дополнения, вычитания, а также относительно операций конкатенации, итерации, гомоморфизма и обращения гомоморфизма.

Теорема *о разрастании для автоматных языков.*

Пусть L — бесконечный автоматный язык. Тогда существует такая константа n, что любое слово $w \in L$ длины $|w| \geq n$ можно разбить на три части x, y и z так, что $w = xyz$ и

1) $|xy| \leq n$;

2) $|y| > 0$;

3) для любого $m \geq 0$ слово $w_m = xy^m z$ принадлежит языку L.

(Здесь $y^0 = \varepsilon,\ y^1 = y,\ y^{i+1} = y^i y$).

Содержательно, эта теорема утверждает, что у всякого достаточно длинного слова из автоматного языка имеется непустое подслово, которое можно вырезать или повторить сколько угодно раз, оставаясь внутри языка.

Для доказательства неавтоматности L можно испольовать схему доказательства “от противного”.

1) Предположим, что L автоматный язык. Тогда для него имеется константа n из утверждения теоремы о разрастании.

2) Определим по n некотрое *“специальное” слово* w из L длины больше n и докажем, что *для любого разбиения* $w = xyz$, удовлетворяющего условиям (1) и (2) теоремы, найдётся такое $k \geq 0$, что слово $w_k = xy^k z$ не принадлежит L.

3) На основании полученного противоречия делаем вывод, что L не автоматный язык.

Примеры неавтоматных языков (из лекции):

1) $L_1 = \{w = 0^i 1^i \mid i \geq 1\}$;

2) СКОБ — язык правильных скобочных последовательностей в алфавите $\{(,)\}$;

3) $L_2 = \{w = 0^i 1^j \mid i \leq 2j + 1\}$;

4) $L_3 = \{|^{i^2} \mid i \geq 0\}$ — язык “квадратов” в унарном алфавите $\{|\}$;

5) $L_{pr} = \{|^p \mid p -$ *простое число* $\}$ — язык “простых чисел” в унарном алфавите $\{|\}$;

6) $L_4 = \{0^i 1^j \mid i \neq j\}$.

Задачи

19.1. Пусть язык L в алфавите $\{a, b, c\}$ состоит из всех слов, которые начинаются на aa и содержат подслово bb. Какая из следующих фраз определяет язык $h(L)$, являющийся образом L при следующем гомоморфизме $h : \{a, b, c\}^* \to \{0, 1\}^*$: $h(a) = 01, h(b) = 11, h(c) = \varepsilon$?

а) Все слова в алфавите $\{0,1\}$, начинающиеся на 0101, с длиной больше 7.
б) Все слова чётной длины в алфавите $\{0,1\}$, начинающиеся на 0101.
в) Все слова чётной длины в алфавите $\{0,1\}$, начинающиеся на 0101, в которых на чётных местах стоят единицы и которые содержат подслово 1111.
г) Все слова в алфавите $\{0,1\}$, начинающиеся на 0101, в которых на чётных местах стоят единицы и которые содержат подслово 1111.

19.2. Пусть язык L в алфавите $\{a,b,c\}$ состоит из всех слов, которые заканчиваются на bcc и содержат подслово aca. Какая из следующих фраз определяет язык $h(L)$, являющийся образом L при следующем гомоморфизме $h:\{a,b,c\}^* \to \{0,1\}^*$: $h(a)=00, h(b)=10, h(c)=\varepsilon$?
а) Все слова в алфавите $\{0,\ 1\}$, заканчивающиеся на 10, с длиной больше 5.
б) Все слова чётной длины в алфавите $\{0,\ 1\}$, содержащие подслово 0000.
в) Все слова чётной длины в алфавите $\{0,\ 1\}$, заканчивающиеся на 10, в которых на чётных местах стоят нули.
г) Все слова в алфавите $\{0,\ 1\}$, заканчивающиеся на 10, в которых на чётных местах стоят нули и которые содержат подслово 0000.
д) Все слова в алфавите $\{0,\ 1\}$, заканчивающиеся на 10, в которых на нечётных местах стоят нули и которые содержат подслово 0000.

19.3. Пусть язык L в алфавите $\{a,b\}$, состоит из всех слов, которые начинаются на aa и содержат число символов a кратное 3, и пусть гомоморфизм $h:\{0,1,2\}^* \to \{a,b\}^*$ задан равенствами $h(0)=aaa, h(1)=ba, h(2)=\varepsilon$. Какие из следующих трёх слов принадлежат прообразу $h^{-1}(L)$ языка L при гомоморфизме h?
а) $w_1 = 21112$;
б) $w_2 = 20101012$;
в) $w_3 = 00211011$.

19.4. Пусть заданы ДКА $A = \langle\{a,b,c\},\{0,1,2,3\},0,F=\{2\},\Phi_A\rangle$ с программой $\Phi_A:\{0a\to 1, 0b\to 1, 0c\to 0, 1a\to 1, 1b\to 2, 1c\to 2, 2a\to 3, 2b\to 3, 2c\to 2, 3a\to 3, 3b\to 3, 3c\to 3\}$ и гомоморфизм $h:\{a,b,c\}^*\to\{0,1\}^*$: $h(a)=01, h(b)=11, h(c)=\varepsilon$. Какие из следующих трёх автоматов C_1, C_2, C_3 распознают гомоморфный образ $h(L_A)$?
$C_1 = \langle\{0,1\},\{0,1,2,3,q0,q1,q2,q3,q4,q5,q6,q7\},0,F_1=\{1,2\},\Phi_1\rangle$,
$C_2 = \langle\{0,1\},\{0,1,2,3,q0,q1,q2\},0,F_2=\{1,2\},\Phi_2\rangle$,

$C_3 = \langle\{0,1\},\{0,1,2,3,q0,q1,q2\},0,F_3 = \{0,1,2\},\Phi_3\rangle$, где программы заданы в следующих таблицах ($\varnothing$ означает отсутствие соответствующего перехода).

Φ_1:

$\Sigma\setminus Q$	0	1	2	3	$q0$	$q1$	$q2$	$q3$	$q4$	$q5$	$q6$	$q7$
0	$q0$	$q2$	$q4$	$q6$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
1	$q1$	$q3$	$q5$	$q7$	1	1	1	2	3	3	3	3

Φ_2:

$\Sigma\setminus Q$	0	1	2	3	$q0$	$q1$	$q2$
0	$q0$	$q1$	3	3	3	3	3
1	$q0$	$q2$	3	3	1	1	2

Φ_3:

$\Sigma\setminus Q$	0	1	2	3	$q0$	$q1$	$q2$
0	$q0$	$q1$	3	3	$q0$	3	3
1	$q0$	$q2$	3	3	1	1	2

19.5. Пусть гомоморфизм $\phi : \{a,b,c\}^* \to \{0,1\}^*$ определяется равенствами

$$\phi(a) = 01,\ \phi(b) = 11,\ \phi(c) = 00.$$

Постройте детерминированный конечный автомат, который распознаёт образ $\phi(L)$ языка

$$L = \{w \mid w \text{ начинается не с } b \text{ и не содержит } aa\}.$$

19.6. Пусть гомоморфизм $\phi : \{0,1\}^* \to \{a,b,c\}^*$ определяется равенствами

$$\phi(0) = aa, \quad \phi(1) = bc.$$

Постройте детерминированный конечный автомат, который распознаёт образ $\phi(L)$ языка

$$L = \{w \mid w \text{ начинается с } 0 \text{ и содержит } 11\}.$$

19.7. Пусть гомоморфизм $\phi : \{a,b,c\}^* \to \{0,1,2\}^*$ определяется равенствами

$$\phi(a) = 10,\ \phi(b) = \varepsilon,\ \phi(c) = 12.$$

Постройте детерминированный конечный автомат, который распознаёт язык $\phi(L)$ для языка

$L = \{w \mid$ буквы c встречаются в w блоками чётной длины и хоть один такой блок имеется$\}$.

19.8. Пусть гомоморфизм $\phi : \{a, b, c\}^* \to \{0, 1\}^*$ определяется равенствами

$$\phi(a) = 10,\ \phi(b) = 01,\ \phi(c) = \varepsilon.$$

Постройте детерминированный конечный автомат, который распознаёт язык $\phi^{-1}(L)$ для языка

$L = \{w \mid w$ заканчивается на 01 и содержит чётное число единиц$\}$.

19.9. Пусть гомоморфизм $\phi : \{a, b, c\}^* \to \{0, 1\}^*$ определяется равенствами

$$\phi(a) = 01,\ \phi(b) = 11,\ \phi(c) = \varepsilon.$$

Постройте детерминированный конечный автомат, который распознаёт прообраз $\phi^{-1}(L)$ языка

$L = \{w \mid w$ начинается с 11 и не содержит 010$\}$.

19.10. Пусть гомоморфизм $\phi : \{a, b, c\}^* \to \{0, 1\}^*$ определяется равенствами

$$\phi(a) = 00,\ \phi(b) = 011,\ \phi(c) = 01.$$

Постройте детерминированный конечный автомат, который распознаёт язык $\phi^{-1}(L)$ для языка

$L = \{w \mid w$ содержит подслово 001 или подслово 10$\}$.

19.11. *Цилиндрификация* — это операция, которая обратна проекции. Для любых алфавитов Δ и Σ таких, что $\Delta \subset \Sigma$, и любого языка L в алфавите Δ определим его цилиндрификацию как язык $CYL_\Sigma(L) = \{w \in \Sigma^* \mid$ при вычёркивании из w всех букв, не входящих в Δ, получается слово $u \in L)$.

Показать, что для автоматного языка L язык $CYL_\Sigma(L)$ также является автоматным языком. Предложите процедуру перестройки автомата, распознающего L, в автомат, распознающий $CYL_\Sigma(L)$.

19.12. Обращением слова $w = w_1 w_2 \ldots w_k$ $(w_i \in \Sigma, i = 1, \ldots, k)$ называется слово $w^{-1} = w_k \ldots w_2 w_1$. Показать, что для автоматного языка L его обращение — язык $L^{-1} = \{w^{-1} \mid w \in L\}$ — также является автоматным языком.

19.13. Пусть L — автоматный язык в алфавите Σ. Доказать, что автоматными являются и следующие языки:

а) ПРЕФ$(L) = \{w \mid$ существует такое слово $x \in \Sigma^*,$ что $wx \in L\}$.

б) СУФ$(L) = \{w \mid$ существует такое слово $x \in \Sigma^*,$ что $xw \in L\}$.

в) КОР$(L) = \{w \mid$ существуют такие слова $x, y \in \Sigma^*,$ что $xwy \in L\}$.

г) $\mathrm{MAX}(L) = \{w \mid w \in L$ и для всякого непустого x слово $wx \notin L\}$.
д) $\mathrm{MIN}(L) = \{w \mid w \in L$ и никакой собственный префикс слова w не принадлежит $L\}$.
е) $\mathrm{DEL}(L) = \{xz \mid$ существует такое слово $y \in \Sigma^*$, что $xyz \in L\}$.
ж) $\mathrm{CYCLE}(L) = \{yx \mid xy \in L\}$.

19.14. Пусть L_1 — автоматный язык в алфавите Σ, а L_2 — произвольный язык в том же алфавите. Докажите, что язык $L_1/L_2 = \{w \in \Sigma^* \mid$ существует такое слово $x \in L_2$, что $wx \in L_1\}$ также является автоматным.

19.15. Тасовкой двух слов одинаковой длины $x = x_1 \dots x_n$ и $y = y_1 \dots y_n$ называется слово $\mathrm{TAC}(x, y) = x_1 y_1 \dots x_n y_n$. Тасовкой двух языков L_1 и L_2 называется язык $\mathrm{TAC}(L_1, L_2) = \{\mathrm{TAC}(x, y) \mid x \in L_1, y \in L_2, |x| = |y|\}$. Докажите, что если языки L_1 и L_2 автоматные, то язык $\mathrm{TAC}(L_1, L_2)$ тоже автоматный.

19.16. Пусть L — автоматный язык в алфавите $\Sigma = \{a_1, \dots, a_m\}$, а $L_1, \dots, L_m$ — это автоматные языки в алфавите Δ. Доказать, что автоматным является и язык $\mathrm{ЗАМ}(L)$, полученный из слов L заменой каждой буквы a_i на некоторое слово из L_i, т.е.
$\mathrm{ЗАМ}(L) = \{w \mid$ существует такое слово $u = a_{i_1} a_{i_2} \dots a_{i_n} \in L$ и такие слова $w_1, w_2, \dots, w_n \in \Delta^*$, что $w = w_1 w_2 \dots w_n$ и $w_j \in L_{i_j}$ для всех $j = 1, 2, \dots n\}$.

19.17. Пусть L — автоматный язык в алфавите Σ, k — целое положительное число и ϕ — отображение Σ^k в Σ. Доказать, что автоматным является язык
$L_1 = \{\phi(a_1 a_2 \dots a_k) \dots \phi(a_{(n-1)k+1} a_{(n-1)k+2} \dots a_{nk}) \mid a_1 a_2 \dots a_{nk} \in L\}$.

19.18. Пусть язык L в алфавите $\{a, b, c\}$, состоит из всех слов, в которых количество букв b превосходит количество букв a не менее чем на 2. Предположим, что L — автоматный язык и что n — это константа, которая существует для него по утверждению теоремы о разрастании. Какое из следующих “специальных” слов позволяет опровергнуть это предположение, т.е. для какого из них не выполнено утверждение (3) теоремы о разрастании?
а) $c^n bbbaaabb$;
б) $ba^n b^{n+4} aaa$;
в) cb^{n+2};
г) $b^{n+2} ca^n c$;
д) $b^n ca^n bbb$.

19.19. Используя теорему о разрастании, установите, какие из следующих языков в алфавите $\{a, b\}$ не являются автоматными.
$L_1 = \{a^2b^na^2 | n > 0\}$;
$L_2 = \{ww | w = a^2b^na^2, n > 0\}$;
$L_3 = \{wv | w = a^2b^na^2, v = b^2a^mb^2$ *для произвольных* $n, m > 0\}$;
$L_4 = \{(01)^i(01)^i \mid i > 0\}$;
$L_5 = \{(01)^i10(01)^i \mid i > 0\}$;
$L_6 = \{0^{[\sqrt{n}]} \mid n \geq 0\}$;
$L_7 = \{0^{|[100\sin(n^2+n)]|} \mid n \geq 0\}$;
$L_8 = \{0^{n^2+n} \mid n \geq 0\}$.

19.20. Докажите, что теорема о разрастании остаётся справедливой и при замене условия 1) $|xy| \leq n$ на условие $1')$ $|yz| \leq n$, т.е. повторяющееся подслово y имеется и в суффиксе w длины $\leq n$.

19.21. Доказать, что следующие языки в алфавите $\Sigma = \{a, b, c\}$ не являются автоматными.
а) Множество всех слов, в которых букв a на 3 больше, чем букв b.
б) $L = \{a^ncb^m \mid m > 3n\}$.
в) $L = \{wcw^{-1} \mid w = a^2b^na$ *для некоторого* $n > 0\}$.
г) $L = \{w \mid |w| = 2^n$ *для некоторого целого числа* $n\}$.
д) $L = \{wc^{|w|} \mid w \in \{a, b\}^*,\ \ |w| -$ *длина слова* $w\}$.

19.22. Пусть λ-выражение — это либо переменная x, либо символ λ, за которым следует переменная, а далее либо λ-выражение, либо левая скобка, λ-выражение, ещё одно λ-выражение и правая скобка. Например, $\lambda xx, \lambda x(xx)$, $\lambda x\lambda x(\lambda x(xx)\lambda x(xx))$ — это правильные λ-выражения, а (xx), $\lambda x(\lambda x)$ и $\lambda x((xx)$ — неправильные. Докажите, что язык λ-выражений в алфавите $\{x, \lambda, (,)\}$ не является автоматным.

19.23. Выше в задаче 17.8 строился автомат-распознаватель, который проверял правильность сложения двоичных чисел. Докажите, что для операции умножения двоичных чисел такого автомата не существует, т.е. что язык в алфавите троек битов
$U = \{(x_1(1), x_2(1), y(1))(x_1(2), x_2(2), y(2)) \dots (x_1(n), x_2(n), y(n)) \mid y = y(n) \dots y(2)y(1)$ — это первые n битов произведения двоичных чисел $x_1 = x_1(n) \dots x_1(2)x_1(1)$ и $x_2 = x_2(n) \dots x_2(2)x_2(1)\}$
не является автоматным.

19.24. Доказать, что язык $L = \{w \mid$ число букв a в $w \neq$ число букв b в $w\}$ в алфавите $\Sigma = \{a, b\}$ не является автоматным.

20. Алгоритмы: структурированные программы

Структурированные программы.

Оператор присваивания имеет один из следующих трёх видов:

а) $x := x + 1$

б) $x := 0$

в) $x := y$.

Условие — это выражение одного из двух видов:

а) $x = y$ или б) $x < y$.

Структурированные программы: синтаксис.

а) Каждое присваивание — это структурированная программа.

б) *Последовательное применение*, или *композиция*, программ:
если Π_1 и Π_2 — структурированные программы, то и
$\Pi = \Pi_1; \Pi_2$ — это структурированная программа.

в) *Условный оператор*:
если Π_1 и Π_2 — структурированные программы, а Φ — это условие, то

$$\Pi = \textbf{если} \;\; \Phi \; \textbf{то} \;\; \Pi_1 \; \textbf{иначе} \;\; \Pi_2 \; \textbf{конец}$$

является структурированной программой.

г) *Оператор цикла*:
если Π_1 — структурированная программа, а Φ — это условие, то

$$\Pi = \textbf{пока} \;\; \Phi \; \textbf{делай} \;\; \Pi_1 \; \textbf{всё}$$

является структурированной программой.

д) Других структурированных программ нет.

Var_Π — множество переменных входящих в программу Π.

Состояние структурированной программы — это отображение σ из множества переменных Var_Π во множество $\mathbf{N}$.

$\sigma(x)$ — значение переменной x в состоянии σ.

$\mathbf{S}$ — множество всех состояний.

Структурированные программы: семантика.

Семантика программы Π — это отбражение (вообще говоря, частичное) типа $\mathbf{S} \to \mathbf{S}$, которое программа Π индуцирует на множестве всех состояний.

$\Pi(\sigma)$ — состояние-результат применения программы Π к состоянию σ. Оно определяется индукцией по построению программы:

а) $(x := x + 1)(\sigma) = \sigma_1$, где $\sigma_1(y) = \sigma(y)$ при $y \neq x$ и $\sigma_1(x) = \sigma(x) + 1$.

б) $(x := 0)(\sigma) = \sigma_1$, где $\sigma_1(y) = \sigma(y)$ при $y \neq x$ и $\sigma_1(x) = 0$.

в) $(x := y)(\sigma) = \sigma_1$, где $\sigma_1(z) = \sigma(z)$ при $z \neq x$ и $\sigma_1(x) = \sigma(y)$

г) Пусть $\Pi = \Pi_1; \Pi_2$. Тогда $\Pi(\sigma) = (\Pi_1; \Pi_2)(\sigma) = \Pi_2(\Pi_1(\sigma))$, при этом, если $\Pi_1(\sigma) = \infty$ или $\sigma_1 = \Pi_1(\sigma)$ и $\Pi_2(\sigma_1) = \infty$, то и $\Pi(\sigma) = \infty$.

д) Пусть $\Pi = \textbf{если} \;\; x = y \; \textbf{то} \;\; \Pi_1 \; \textbf{иначе} \;\; \Pi_2 \; \textbf{конец}$. Тогда

$$\Pi(\sigma) = \begin{cases} \Pi_1(\sigma), & \textit{если } \Pi_1(\sigma) < \infty \ \textit{ и } \ \sigma(x) = \sigma(y), \\ \Pi_2(\sigma), & \textit{если } \Pi_2(\sigma) < \infty \ \textit{ и } \ \sigma(x) \neq \sigma(y), \\ \infty & \textit{в остальных случаях.} \end{cases}$$

е)Пусть $\Pi =$ **если** $x < y$ **то** Π_1 **иначе** Π_2 **конец**. Тогда

$$\Pi(\sigma) = \begin{cases} \Pi_1(\sigma), & \textit{если } \Pi_1(\sigma) < \infty \ \textit{ и } \ \sigma(x) < \sigma(y), \\ \Pi_2(\sigma), & \textit{если } \Pi_2(\sigma) < \infty \ \textit{ и } \ \sigma(x) \geq \sigma(y), \\ \infty & \textit{в остальных случаях.} \end{cases}$$

ж) Пусть $\Pi =$ **пока** $x = y$ **делай** Π_1 **всё**. Тогда при $\sigma(x) \neq \sigma(y)$ $\Pi(\sigma) = \sigma$, а при $\sigma(x) = \sigma(y)$ $\Pi(\sigma)$ — это первое такое состояние σ_m в последовательности состояний $\sigma_0 = \sigma, \sigma_1 = \Pi(\sigma_0), \ldots, \sigma_{i+1} = \Pi(\sigma_i), \ldots$, что при $i \leq m$ все состояния σ_i определены, при $i < m$ имеет место $\sigma_i(x) = \sigma_i(y)$, и $\sigma_m(x) \neq \sigma_m(y)$.
з) Семантику для цикла с условием $x < y$ определите самостоятельно (см. задачу 20.2).

Функция, вычислимая структурированной программой.
Программа Π с входными переменными $x_1, \ldots, x_n$ и результирующей переменной y вычисляет частичную функцию $F : \mathbf{N}^n \to \mathbf{N}$, если для любого набора значений аргументов $a_1, \ldots, a_n$ $(a_i \in \mathbf{N})$ она переводит начальное состояние σ, в котором $\sigma(x_i) = a_i$ при $1 \leq i \leq n$ и $\sigma(z) = 0$ при $z \in Var \setminus \{x_1, \ldots, x_n\}$, в состояние $\sigma_1 = \Pi(\sigma)$ тогда и только тогда, когда $(a_1, \ldots, a_n) \in \delta_F$ и $F(a_1, \ldots, a_n) = \sigma_1(y)$.
$\Phi_{\Pi,y}(x_1, \ldots, x_n)$ — функция, вычислимая программой Π с входными переменными $x_1, \ldots, x_n$ в (результирующей) переменной y.
Арифметическая функция $F(x_1, \ldots, x_n)$ *программно вычислима*, если существует программа Π, для которой $\Phi_{\Pi,y}(x_1, \ldots, x_n) = F(x_1, \ldots, x_n)$.

Задачи

20.1. Какие из следующих последовательностей операторов являются синтаксически правильными структурированными программами?
P1: $x := x + 1;\ z := y;$ если $x + 1 < z$ то $y := z$ иначе $y := x$ конец
P2: $x := x + 1;\ z := y; z := z + 1;$ если $x = z$ то $y := z$ иначе $y := x$ конец
P3: $x := y;\ u := z;\ u := u + 1;$ пока $u > z$ делай $y := z;\ u := u + 1$ всё
P4: $x := z + 1;\ z := y;$ если $x + 1 < z$ то $y := z$ иначе $x := y$ конец
P5: $x := x;\ x := x;\ y := y;\ y := y$
P6: $x := y;\ u := z;\ u := u + 1;$ пока $z < u$ делай $y := z; u := u + 1$ всё; $x := y;$
P7: $x := x + 1;\ z := y; z := z + 1;$ если $x = z$ то $y := z;\ z := z + 1;$ иначе $y := x$ конец

20.2. Определите семантику для программ вида
$\Pi =$ **пока** $x < y$ **делай** Π_1 **всё**.

20.3. Пусть структурированная программа
P: $x := y;\ x := x+1;\ y := z;\ y := y+1;\ z := z+1;\ y := y+1;\ z := y;\ z := z+1;\ x := x+1$
начинает работу в состоянии $\sigma : \sigma(x) = 3, \sigma(y) = 4, \sigma(z) = 2$. В каком состоянии она завершит свою работу?

20.4. Пусть структурированная программа **P**:

$x := y;\ x := x+1;\ y := u;\ y := y+1;\ v := z;\ v := v+1;$
если $x < v$ **то**
  если $x = y$ **то**
    $z := y;\ z := z+1$
  иначе
    $z := x$
  конец
иначе
  $z := x+1$
конец

начинает работу в состоянии $\sigma : \sigma(x) = 0, \sigma(y) = 3, \sigma(z) = 5, \sigma(u) = 4, \sigma(v) = 2$. В каком состоянии она завершит свою работу?

20.5. Пусть структурированная программа **P**:

$x := y;\ x := x+1;\ v := u;\ v := v+1;$
пока $x < v$ **делай**
  если $y < x$ **то**
    $y := y+1$
  иначе
    $x := x+1;\ u := u+1$
  конец
всё

начинает работу в состоянии $\sigma : \sigma(x) = 2, \sigma(y) = 3, \sigma(u) = 5, \sigma(v) = 0$. В каком состоянии она завершит свою работу?

20.6. Пусть Π_+ — это программа, которая вычисляет функцию $\Phi_+(x, y) = x + y$ в переменной x, не изменяя y и используя одну рабочую переменную z. Какие из следующих структурированных программ

Π_1, Π_2, Π_3 вычисляют в переменной x произведение $x * y$?

Π_1	Π_2	Π_3
$x1 := x; i := 0;$ $x := 0;$ **пока** $i < x1$ **делай** $\Pi_+;$ $z := 0;\ i := i + 1$ **всё**	**если** $y < x$ **то** $u := y;\ y := x$ **иначе** $u := x$ **конец**; $x := 0;$ **пока** $i < u$ **делай** $z := 0;\ \Pi_+;\ i := i + 1$ **всё**	**если** $y = 0$ **то** $x := 0$ **иначе** **пока** $i < y$ **делай** $\Pi_+;$ $z := 0;\ i := i + 1$ **всё** **конец**

20.7. Пусть $\Pi_\times$ — это программа, которая вычисляет функцию $\Phi_\times(x, y) = x * y$ в переменной x, используя две рабочие переменные z и i (см. задачу 20.6). Какие из следующих структурированных программ Π_1, Π_2, Π_3 вычисляют в переменной x квадратный корень из x, т.е. функцию $sqrt(x) = \lfloor\sqrt{x}\rfloor$?

Π_1	Π_2	Π_3
$u := x;\ x := 0;\ y := 0;$ **пока** $x < u$ **делай** $z := 0; i := 0;$ $\Pi_\times;$ $y1 := y; y := y + 1;$ $x := y$ **всё**; **если** $x = u$ **то** $x := y$ **иначе** $x := y1$ **конец**	$u := x;\ x := 0;\ y := 0;$ **пока** $x < u$ **делай** $y1 := y;\ y := y + 1;$ $x := y;$ $\Pi_\times;\ z := 0;\ i := 0$ **всё**; **если** $x = u$ **то** $x := y$ **иначе** $x := y1$ **конец**	$u := x;\ u := u + 1; x := 0;$ $y := 0;$ **пока** $x < u$ **делай** $y1 := y; y := y + 1;$ $x := y;$ $\Pi_\times;\ z := 0;\ i := 0$ **всё**; $x := y1$

20.8. Пусть $\Pi_\times$ — это программа, которая вычисляет функцию $\Phi_\times(x, y) = x * y$ в переменной x, используя две рабочие переменные z и i (см. задачу 20.6). Какие из следующих структурированных программ Π_1, Π_2, Π_3 вычисляют в переменной x целую часть частного $\lfloor x/y\rfloor$ (пусть при $y = 0$ результат равен 0)?

Π_1	Π_2	Π_3
$u := x;\ u := u+1;\ x := 0;$ **пока** $x < u$ **делай** $j1 := j;\ j := j+1;$ $\Pi_\times;\ z := 0;\ i := 0$ **всё**; $x := j1$	**если** $y = 0$ **то** $x := 0$ **иначе** $u := x;\ x := 0;\ j := 0;$ **пока** $x < u$ **делай** $j1 := j;\ j := j+1;$ $x := j;$ $\Pi_\times;\ z := 0;\ i := 0$ **всё**; **если** $x = u$ **то** $x := j$ **иначе** $x := j1$ **конец** **конец**	**если** $y = 0$ **то** $x := 0$ **иначе** $u := x;\ x := 0;$ **пока** $x < u$ **делай** $z := 0; i := 0;$ $\Pi_\times;$ $j1 := j;\ j := j+1;$ $x := j;$ **всё**; **если** $x = u$ **то** $x := j$ **иначе** $x := j1$ **конец** **конец**

20.9. Построить структурированные программы, вычисляющие в z следующие функции, и доказать их корректность:
а) $f_\times(x,y) = x * y$ (см. задачу 20.6);
б) $f_{fact}(x) = x!$;
в) $f_{-1}(x) = x \dot{-} 1$, где $0 \dot{-} 1 = 0$ и $(x+1) \dot{-} 1 = x$;
г) $f_-(x,y) = x \dot{-} y$, где $x \dot{-} y = x - y$, если $x \geq y$, и $x \dot{-} y = 0$, если $x < y$;
д) $f_{sqr}(x) = \lfloor \sqrt{x} \rfloor$ (см. задачу 20.7);
е) $f_{exp}(x) = 2^x$;
ж) $f_{log}(x) = \lfloor \log_2 x \rfloor$;
з) $f_/(x,y) = \lfloor x/y \rfloor$ (см. задачу 20.8);
и) $f_{mod}(x,y) = x \mod y$ (остаток от деления x на y);
к) $f_d(x) =$ число различных делителей числа x (пусть $f_d(0) = 0$).

20.10. Пусть Π — структурированная программа и $|Var_\Pi| = m$. Из определений следует, что при различной фиксации входных переменных и выходной переменной программа может вычислять различные функции.
а) Каково максимальное число функций от $n \leq m$ переменных, которое может вычислять Π? Сколько всего разных функций может вычислить Π?
б) Постройте программу $\Pi(m,n)$, которая вычисляет максимальное число различных функций от $n \leq m$ переменных.
в) Постройте программу $\Pi(m)$ с $|Var_\Pi| = m$, которая для каждого $n \leq m$ вычисляет максимальное число различных функций от n переменных.

20.11. Построить структурированные программы, вычисляющие в z следующие функции:

$$\text{а)} \qquad f_1(x,y) = \begin{cases} \lfloor (x+y)/2 \rfloor, & \text{если } x < y, \\ x^2y^3 & \text{в противном случае.} \end{cases}$$

$$\text{б)} \qquad f_2(x,y) = \begin{cases} 3^y, & \text{если } \log_2(x+1) \geq y, \\ |x-y| & \text{в противном случае.} \end{cases}$$

$$\text{в)} \qquad f_3(x,y) = \begin{cases} \lfloor x * y/2 \rfloor, & \text{если } \log_2 x \leq y+2, \\ x^{\lfloor y/2 \rfloor} & \text{в противном случае.} \end{cases}$$

$$\text{г)} \qquad f_4(x,y) = \begin{cases} \lfloor \log_3(x+y+1) \rfloor, & \text{если } 2x \leq y3y, \\ \lfloor \sqrt{x+y} \rfloor & \text{в противном случае.} \end{cases}$$

$$\text{д)} \qquad p(x) = \begin{cases} 1, & \text{если } x \text{ — простое число,} \\ 0 & \text{в противном случае.} \end{cases}$$

20.12. Пусть имеются операции вида $x := 0$, $x := y$, $x := x + 1$, а также ветвление и цикл с условиями вида $x = y$. Напишите программу для проверки условия $x < y$.

20.13. Пусть структурированная программа Π с переменными $Var_\Pi = \{x, y, z_1, \ldots, z_m\}$ вычисляет в y некоторую всюду определенную взаимно однозначную функцию $f(x)$, область значений которой совпадает с множеством всех натуральных чисел $\mathbf{N}$. Тогда обратное отображение $f^{-1}(x) = \{z \mid f(z) = x\}$ также является взаимно однозначной функцией. Постройте структурированную программу, которая вычисляет $f^{-1}(x)$.

20.14. В задаче 2.11 были определены числа Фибоначчи: $F(0) = 0$, $F(1) = 1, \ldots, F(x+2) = F(x) + F(x+1)$.

Постройте структурированную программу, которая вычисляет функцию $F(x)$.

20.15. Для каждой из заданных ниже рекурсивными соотношениями функций $A(x,y)$, $B(x,y)$, $C(x,y)$, $D(x,y)$, $E(x,y)$, $G(x,y)$ и $H(x,y)$ построить вычисляющую её структурированную программу.

а) $A(x,y)$:
$A(0,y) = y + 1$,
$A(x+1, 0) = 2A(x, 1)$,

$A(x+1, y+1) = A(x, y) + xy.$

б) $B(x, y):$
$B(0, y) = 2y,$
$B(x+1, 0) = B(x, 1) + 1,$
$B(x+1, y+1) = x + B(x+1, y).$

в) $C(x, y):$
$C(0, y) = y + 2,$
$C(x+1, 0) = C(x, 1) + 1,$
$C(x+1, y+1) = C(x+1, y) \dot{-} x.$

г) $D(x, y):$
$D(0, y) = y + 1,$
$D(x+1, y+1) = D(x, y+1) + x^y.$

д) $E(x, y):$
$E(0, y) = y + 1,$
$E(x+1, 0) = E(x, 1),$
$E(x+1, y+1) = xE(x+1, y).$

е) $G(x, y):$
$G(0, 0) = 3,$
$G(x+1, 0) = G(x, 0) + \lfloor \log_2(x+1) \rfloor,$
$G(x+1, y+1) = G(x+1, y) + x^2 y.$

ж) $H(x, y):$
$H(0, 0) = 0,$
$H(x+1, 0) = H(x, 1) + 2,$
$H(x+1, 1) = H(x, 0) + 1,$
$H(x+1, y+2) = y + H(x+1, y).$

20.16. Докажите, что никакая из функций $f_+(x, y) = x + y$, $f_-(x, y) = x \dot{-} y$, $f_\times(x, y) = x * y$ и $f_/(x, y) = \lfloor x/y \rfloor$ не вычисляется никакой структурированной программой без циклов.

21. Алгоритмы: частично рекурсивные функции

Оператор суперпозиции.
Пусть F^m и $f_1^n, \ldots, f_m^n$ — арифметические функции. Скажем, что функция G^n получена из $F^m, f_1^n, \ldots, f_m^n$ с помощью оператора *суперпозиции* (обозначение: $G^n = [F^m; f_1^n, \ldots, f_m^n]$), если для всех наборов аргументов $(x_1, \ldots, x_n)$

$$G^n(x_1, \ldots, x_n) = F^m(f_1^n(x_1, \ldots, x_n), \ldots, f_m^n(x_1, \ldots, x_n)).$$

При этом для каждого набора аргументов $(a_1, \ldots, a_n)$ функция $G^n(a_1, \ldots, a_n) < \infty$ (т.е. определена), если определены все значения $f_1^n(a_1, \ldots, a_n) = b_1, \ldots, f_m^n(a_1, \ldots, a_n) = b_m$ и $F^m(b_1, \ldots, b_m) < \infty$.
Оператор примитивной рекурсии.
Скажем, что функция $F^{n+1}(x_1, \ldots, x_n, y)$ получена с помощью оператора *примитивной рекурсии* из функций $g^n(x_1, \ldots, x_n)$ и $h^{n+2}(x_1, \ldots, x_n, y, z)$, если она может быть задана схемой примитивной рекурсии

$$\begin{cases} F^{n+1}(x_1, \ldots, x_n, 0) = g^n(x_1, \ldots, x_n), \\ F^{n+1}(x_1, \ldots, x_n, y+1) = h^{n+2}(x_1, \ldots, x_n, y, F^{n+1}(x_1, \ldots, x_n, y)). \end{cases}$$

В этом случае будем писать $F^{n+1} = R(g^n, h^{n+2})$.
В случае, когда $n = 0$, т.е. F зависит от одного аргумента y, а аргументов $x_1, \ldots, x_n$ нет, схема примитивной рекурсии принимает вид

$$\begin{cases} F^1(0) = a, \\ F^1(y+1) = h^2(y, F^1(y)), \end{cases}$$

где $a \in \mathbf{N}$.
Оператор минимизации.
Функция $F^n(x_1, \ldots, x_n)$ получена с помощью оператора *минимизации* (μ-оператора) из функции $g^{n+1}(x_1, \ldots, x_n, y)$, если $F^n(x_1, \ldots, x_n)$ определена и равна y тогда и только тогда, когда все значения $g^{n+1}(x_1, \ldots, x_n, 0), \ldots, g^{n+1}(x_1, \ldots, x_n, y-1)$ определены и не равны 0, а $g^{n+1}(x_1, \ldots, x_n, y) = 0$. В этом случае будем писать

$$F^n(x_1, \ldots, x_n) = \mu y[g^{n+1}(x_1, \ldots, x_n, y) = 0].$$

Простейшие рекурсивные функции.
Функция называется *простейшей*, если она является одной из следующих функций:
а) $o^1(x) = 0$ — тождественный нуль;
б) $s^1(x) = x + 1$ — следующее число (плюс один);
в) функцией выбора аргумента вида $I_m^n(x_1, \ldots, x_n) = x_m$ $(1 \leq m \leq n)$.

Частично рекурсивные функции.
f — *частично рекурсивная функция (ч.р.ф.),* если она является одной из простейших функций или может получиться из них с помощью *конечного числа* применений операторов суперпозиции, примитивной рекурсии и минимизации, т.е. существует последовательность функций $f_1, f_2, \ldots, f_n = f$, каждая из которых либо является простейшей, либо получена из предыдущих с помощью одного из указанных операторов. Указанная последовательность функций называется *частично рекурсивным описанием функции* f.
Функция f называется *общерекурсивной функцией (о.р.ф.),* если она частично рекурсивна и всюду определена.
Функция f называется *примитивно рекурсивной функцией (п.р.ф.),* если она частично рекурсивна и для неё существует частично рекурсивное описание, использующее лишь операторы суперпозиции и примитивной рекурсии. В таком случае оно называется *примитивно рекурсивным описанием функции* f.
Примеры.
Сложение: $+^2(x, y) = x + y$.

$$\begin{cases} x + 0 = x = I_1^1(x), \\ x + (y+1) = (x+y) + 1 = [s^1; (x+y)] = [s^1;\ I_3^3(x, y, x+y)]. \end{cases}$$

Следовательно, $+^2 = R(I_1^1, [s^1; I_3^3])$.
Умножение: $\times^2(x, y) = x \times y$.

$$\begin{cases} x \times 0 = 0 = o^1(x), \\ x \times (y+1) = (x \times y) + x = I_3^3(x, y, x \times y) + I_1^3(x, y, x \times y). \end{cases}$$

Следовательно, $\times^2 = R(o^1, [+; I_3^3, I_1^3])$.
Минус 1: $\dot{-}1(0) = 0,\ \dot{-}1(x+1) = x$.
$\dot{-}1 = R(0, I_1^2)$.
Вычитание : $x \dot{-} y = x - y$, если $x \geq y$ и $x \dot{-} y = 0$, если $x < y$.

$$\begin{cases} x \dot{-} 0 = x = I_1^1(x), \\ x \dot{-} (y+1) = (x \dot{-} y) \dot{-} 1 = [\dot{-}1^1; I_3^3(x, y, x \dot{-} y)]. \end{cases}$$

Следовательно, $\dot{-}^2 = R(I_1^1, [\dot{-}1^1; I_3^3])$.
Предикаты равенства и неравенства нулю:

$$sg(x) = \begin{cases} 0, & \text{если } x = 0, \\ 1, & \text{если } x \neq 0 \end{cases} \qquad \overline{sg}(x) = \begin{cases} 1, & \text{если } x = 0, \\ 0, & \text{если } x \neq 0 \end{cases}$$

Примитивная рекурсивность этих функций следует из равенств
$sg = R(0, [s^1; [o^1; I_1^2]])$ и $\overline{sg}(x) = 1 \dot{-} sg(x)$.

Суммирование и произведение.
Пусть функция $f(x_1,\ldots,x_n,y)$ является частично (примитивно) рекурсивной. Тогда и функции F^{n+1} и G^{n+1}, заданные следующими равенствами
$F(x_1,\ldots,x_n,y)=\sum\limits_{i=0}^{y} f(x_1,\ldots,x_n,i)$,
$G(x_1,\ldots,x_n,y)=\prod\limits_{i=0}^{y} f(x_1,\ldots,x_n,i)$,
являются частично (примитивно) рекурсивными.

Ограниченная минимизация. Пусть примитивно рекурсивная функция $g(x,y)$ такова, что для каждого x найдется $y\le x$, для которого $g(x,y)=0$. Тогда функция $F(x)=\mu y[g(x,y)=0]$ примитивна рекурсивна. Будем обозначать эту функцию через $\mu y\le x[g(x,y)=0]$.

Задачи

21.1. Пусть заданы три функции: $f(x,y,z)=xy+z$, $g(x,y)=2x+y$, $h(x)=2x^2$. Какую функцию $F(x_1,x_2)$ задаёт выражение $[f;[h;I_1^2],[g;[h;I_2^2],I_2^2],I_2^2]$?

21.2. Определите, какие функции задают следующие выражения.
а) $R(0,[+;[s^1;[+;I_1^2,I_1^2]],I_2^2])$
б) $R(0,[+;[+;[s^1;I_1^2],[s^1;I_1^2]],I_2^2])$
в) $R(0,[+;[+;[s^1;I_1^2],I_1^2],I_2^2])$

21.3. Функция $f(x)$ задана выражением $R(0,[+;[+;[+;[s^1;I_1^2],[s^1;I_1^2]],[+;I_1^2,I_1^2]],[+;I_2^2,I_2^2]])$. Докажите, что $f(x)=3\cdot 2^{x+1}-4x-6$.

21.4. а) Пусть функция $F(x)$ задана примитивной рекурсией $R(1,h(y,z))$, где $h(y,z)=\lfloor 2^z/z^2\rfloor$. Чему равно значение $F(5)$?
б) Пусть функция $F(x)$ задана примитивной рекурсией $R(1,h(y,z))$, где $h(y,z)=\lfloor 2^z/z\rfloor$. Чему равно значение $F(5)$?
в) Пусть функция $F(x)$ задана примитивной рекурсией $R(2,h(y,z))$, где $h(y,z)=z^2-2y-1$. Чему равно значение $F(4)$?

21.5. Для какой из следующих функций $f(x,y)$ выражение $\mu y[f(x,y)=0]$ задаёт функцию $F(x)=\lfloor\sqrt{x}\rfloor$ (целая часть квадратного корня из x)?
а) $f(x,y)=y^2\dot{-}x$,
б) $f(x,y)=x\dot{-}y^2$,
в) $f(x,y)=(x+1)\dot{-}y^2$,

г) $f(x,y) = (x+1)\dot{-}(y+1)^2)$,
д) $f(x,y) = x\dot{-}(y+1)^2)$.

21.6. Для какой из следующих функций $f(x,y,i)$ выражение $\mu i[f(x,y,i) = 0]$ задаёт функцию $F(x,y) = \lfloor y/x \rfloor$ (целая часть частного от деления y на x)?
а) $f(x,y.i) = y\dot{-}ix$,
б) $f(x,y,i) = (y+1)\dot{-}i(x+1))$,
в) $f(x,y,i) = y\dot{-}(i+1)x$,
г) $f(x,y,i) = (y+1)\dot{-}ix$,
д) $f(x,y,i) = (y+1)\dot{-}(i+1)x$.

21.7. Пусть функция $rm(x,y) = y \textbf{ mod } x$ равна остатку от деления y на x (положим $rm(0,y) = y$). Какое из следующих выражений определяет число $dn(x)$ различных делителей числа x, отличных от 1 и самого x?
а) $x - \sum\limits_{i=0}^{x} sg(rm(i,x))$
б) $\sum\limits_{i=0}^{x}(1 - sg(rm(i,x)))$
в) $\sum\limits_{i=0}^{x} sg(rm(i,x)) - 2$
г) $\sum\limits_{i=0}^{x}(1 - sg(rm(i,x))) - 2$

21.8. Показать, что следующие функции являются примитивно рекурсивными.
а) $exp(x,y) = x^y$;
б) $fact(x) = x!$;
в) $min(x,y) =$ наименьшее из x и y;
г) $max(x,y) =$ наибольшее из x и y;
д) $div(x,y) =$ целая часть частного от деления y на x (пусть $div(0,y) = y$);
е) предикаты равенства и неравенства:

$$eq(x,y) = \begin{cases} 1, & \text{если } x = y, \\ 0, & \text{если } x \neq y, \end{cases} \qquad \overline{eq}(x,y) = \begin{cases} 0, & \text{если } x = y, \\ 1, & \text{если } x \neq y; \end{cases}$$

ж) $f(x) = 2^{2^x}$.

21.9. Доказать, что следующая функция является примитивно рекурсивной:

$$f(x,y) = \begin{cases} \lfloor \log_2 \log_2(x+y+1) \rfloor, & \text{если } 2x \leq 3y \\ x^2 + 3y & \text{в противном случае.} \end{cases}$$

21.10. Доказать, что следующая функция является примитивно рекурсивной:

$$f(x,y) = \begin{cases} \lfloor\sqrt{x+y}\,\rfloor, & \text{если } \log_3 x \le y^2 \\ 2^{x+y} & \text{в противном случае.} \end{cases}$$

21.11. Докажите, что если $f(x_1,\dots,x_n)$ является ч.р.ф. (п.р.ф.), то и функция
$g(x_1,\dots,x_n) = f(x_{i_1},\dots,x_{i_n})$ является ч.р.ф. (п.р.ф.) для любой перестановки
$(i_1,\dots,i_n)$ чисел $1,2,\dots,n$.

21.12. Оператор сдвига.
Пусть $g(x_1,\dots,x_n)$ — частично (примитивно) рекурсивная функция, a и $b>0$ — числа из $\mathbf{N}$. Тогда и функция

$$f(x_1,\dots,x_n) = \begin{cases} a, & \text{если } \ x_n \le b, \\ g(x_1,\dots,x_n-b), & \text{если } \ x_n > b \end{cases}$$

является частично (примитивно) рекурсивной.

21.13. Показать, что следующие функции являются частично (примитивно) рекурсивными.
а) $rt(n,x) = \lfloor\sqrt[n]{x}\rfloor$ — корень n-й степени из x (целая часть);
б) $log(i,x) = \lfloor\log_i x\rfloor$ (пусть при $i\in\{0,1\}$ или $x=0$ $\quad log(i,x)=0$);
в) $t(x)$ = число различных делителей числа x ($t(0)=0$);
г) $p(x)=1$, если x — простое число, и $p(x)=0$, если x составное;
д) $pn(k)$ — k-е простое число в порядке возрастания ($pn(0)=0, pn(1)=2, pn(2)=3, pn(3)=5,\dots$);
е) $d(n,m,i)$ — i-й знак в m-ичном разложении числа n, т.е. если $n=\sum\limits_{i=0}^{\infty} a_i m^i$, где $0\le a_i\le m-1$, то $d(n,m,i)=a_i$;
ж) $nod(x,y)$ = наибольший общий делитель чисел x и y (пусть $nod(0,y)=nod(x,0)=0$).

21.14. Докажите, что если значения общерекурсивной функции $f(x)$ изменить на конечном множестве, то получившаяся функция $f'(x)$ также будет общерекурсивной.

21.15. Доказать, что из функции $o(x)=0$ и из функций выбора $I^n_m(x_1,...,x_n)=x_m$ с помощью суперпозиции и примитивной рекурсии нельзя получить функцию $s(x)=x+1$ и функцию $d(x)=2*x$.

21.16. Пусть функция $g(x_1, ..., x_n, y)$ примитивно рекурсивна. Доказать, что функция

$$f(x_1, ..., x_n, y, z) = \begin{cases} \sum\limits_{i=0}^{z} g(x_1, ...x_n, y+i) & \text{при } y \leq z, \\ 0 & \text{при } \ y > z \end{cases}$$

примитивно рекурсивна.

21.17. Доказать, что если функции $f(x_1, ..., x_n, y), g(x_1, ..., x_n, y)$ и $h(x_1, ..., x_n, y)$ общерекурсивны, то функция
$F(x_1, ..., x_n) = min\{ \ y \ | \ f(x_1, ..., x_n, y) = 0$ или $g(x_1, ..., x_n, y) > h(x_1, ..., x_n, y)\}$
является частично рекурсивной.

21.18. Определим для любой пары чисел (x, y) её номер $c_2(x, y) = 2^x(2y+1) - 1$ и "обратные" координатные функции:
$c_{21}(z) =$ *максимальная степень 2, на которую делится* $z+1$ и
$c_{22}(z) = [($*максимальное нечётное число, на которое делится* $z+1)\dot{-}1]/2$.
Определим теперь по индукции функции c_n нумерации n-ок чисел при $n > 2$ и обратные им координатные функции c_{ni} $(1 \leq i \leq n)$:
$c_n(x_1, x_2, x_3, \ldots, x_n) = c_{n-1}(c_2(x_1, x_2), x_3, \ldots, x_n)$,
$c_{n1}(z) = c_{21}(c_{(n-1)1}(z))$,
$c_{n2}(z) = c_{22}(c_{(n-1)1}(z))$,
$c_{n(i+2)}(z) = c_{(n-1)(i+1)}(z) \qquad (i = 1, \ldots, n-2)$.
а) Докажите, что для каждого $n \geq 2$ и для любого z имеет место равенство $c_n(c_{n1}(z), c_{n2}(z), \ldots, c_{nn}(z)) = z$. (Проверьте это свойство индукцией по n.)
б) Докажите, что для каждого $n \geq 2$ функция нумерации n-ок $c_n(x_1, \ldots, x_n)$ и обратные ей функции выбора i-го элемента набора $c_{ni}(z)$ $(1 \leq i \leq n)$ являются примитивно рекурсивными.

21.19. а) Найдите для нумерации n-ок, задаваемой $c_n(x, y)$, номера последовательностей $(1, 3)$, $(3, 1)$, $(1, 2, 3)$, $(0, 4, 1)$.
б) Найдите пары с номерами 9 и 11 и тройки с номерами 351 и 90111.
в) Найдите номера последовательностей $(0, \ldots, 0, x)$ и $(1, 0, \ldots, 0)$ (первая последовательность содержит $n-1$ нулей и заканчивается на x, а вторая последовательность содержит единицу, а далее $n-1$ нулей).

21.20. Предположим, что все пары (x, y) натуральных чисел упорядочены по возрастанию суммы $(x+y)$, а внутри группы пар с одинаковой суммой — по возрастанию x-координаты. Этот порядок выглядит так: $(0,0),(0,1),(1,0),(0,2),(1,1),(2,0),\ldots,(0,x+y),(1,x+y-1),\ldots,(x,y),\ldots,(x+y,0),\ldots$.

Пусть $d(x,y)$ — это номер пары (x,y) в этом порядке (будем считать, что пара (0,0) имеет номер 0). Тогда функция d^2 однозначно нумерует все пары.

а) Докажите, что $d(x,y)=\frac{(x+y)(x+y+1)}{2}+x$.

б) Найдите обратные функции $d_1(z)$ и $d_2(z)$ такие, что $d_1(d(x,y))=x$, $d_2(d(x,y))=y$ и, следовательно, $d(d_1(z),d_2(z))=z$.

21.21. Пусть $c_2(x,y)$ — это функция нумерации пар, а $c_{21}(z)$ и $c_{22}(z)$ — это соответствующие обратные функции такие, что $c_2(c_{21}(z),c_{22}(z))=z$ для всех z (такие функции определены, например, в задачах 21.18 и 21.20). Примитивную рекурсивность этих функций можно использовать для установления рекурсивности функций, значения которых на аргументе $(y+1)$ зависят от их значений в двух предыдущих точках $y-1$ и y.

а) Пусть функция $F(y)$ (числа Фибоначчи) задана равенствами

$$F(0)=0, F(1)=1,\ F(y+2)=F(y)+F(y+1).$$

Положим $G(y)=c_2(F(y),F(y+1))$. Так как $F(y)=c_{21}(G(y))$, то для доказательства примитивной рекурсивности F достаточно установить примитивную рекурсивность G. Определите, какая из следующих примитивных рекурсий задает G.

Ответ 1. $R(2,[c_2;[c_{21};I^2_2],[+;[c_{21};I^2_2],[c_{22};I^2_2]]])$

Ответ 2. $R(1,[c_2;[c_{22};I^2_2],[+;[c_{21};I^2_1],[c_{22};I^2_2]]])$

Ответ 3. $R(2,[c_2;[c_{22};I^2_2],[+;[c_{21};I^2_2],[c_{22};I^2_2]]])$

Ответ 4. $R(5,[c_2;[c_{22};I^2_2],[+;[c_{21};I^2_2],[c_{22};I^2_2]]])$

Ответ 5. Ни одна из выше перечисленных.

б) Пусть функция $G(y)$, задана равенствами

$G(0)=1$,

$G(1)=2$,

$G(y+2)=G(y)*G(y+1)$.

Положим $H(y)=c_2(G(y),G(y+1))$.

Определите, какая из следующих примитивных рекурсий задает H.

Ответ 1. $R(9,[c_2;[c_{21};I^2_2],[\times;[c_{21};I^2_2],[c_{22};I^2_2]]])$

Ответ 2. $R(9,[c_2;[c_{22};I^2_2],[\times;[c_{21};I^2_2],[c_{22};I^2_2]]])$

Ответ 3. $R(2,[c_2;[c_{22};I^2_2],[\times;[c_{21};I^2_1],[c_{22};I^2_2]]])$

Ответ 4. $R(2,[c_2;[c_{22};I^2_2],[\times;[c_{21};I^2_1],[c_{22};I^2_2]]])$

Ответ 5. Ни одна из выше перечисленных.

22. Алгоритмы: машины Тьюринга

Машина Тьюринга — это система вида

$$\mathcal{M} = \langle Q, \Sigma, P, q_0, q_f\rangle,$$

включающая следующие компоненты:
$Q = \{q_0, q_1, \ldots, q_n\}$ — *алфавит состояний*;
$\Sigma = \{a_0, a_1, \ldots, a_{m-1}\}$ — *алфавит ленты*;
P — *программа машины*, в которой для каждой пары $q_i \in Q \backslash \{q_f\}$, $a_j \in \Sigma$ имеется (одна!) команда вида $q_i a_j \to q_k a_l C$, которая означает, что *машина, находясь в состоянии q_i и наблюдая символ a_j, в следующий момент перейдёт в состояние $q_k \in Q$, заменит наблюдаемый символ на $a_l \in \Sigma$ и сдвинет головку в направлении $C \in$ {Л, П, Н} (Л задает сдвиг головки влево, П — вправо, Н — на месте)*;
$q_0 \in Q$ — *начальное* состояние;
$q_f \in Q$ — *заключительное* состояние.
Выделим в алфавите Σ специальный *пустой* символ $a_0 = \wedge$ и будем считать, что во всех ячейках ленты, кроме конечного их числа, в начальный и во все последующие моменты находится пустой символ.
Конфигурация машины Тьюринга м.Т. $\mathcal{M}$ в некоторый момент времени — слово $K = w_л q_i a_j w_п$, где $w_л \in \Sigma^*$ — слово на ленте левее текушего положения головки, q_i — внутреннее состояние в данный момент, a_j - символ, обозреваемый головкой, $w_п \in \Sigma^*$ — слово на ленте правее текушего положения головки.
Начальная конфигурация — это конфигурация вида $q_0 w$, т.е. в начальный момент времени головка в состоянии q_0 обозревает первый символ входного слова w.
Заключительная конфигурация — это конфигурация вида $w_1 q_f w_2$, в которой машина находится в заключительном состоянии q_f.
Скажем, что конфигурация $K = w_1 q_i a_j w_2$ м.Т. $\mathcal{M}$ *за один шаг (такт) переходит в конфигурацию* $K' = w_1' q_k a_{j'} w_2'$ ($K \vdash_{\mathcal{M}} K'$), если в программе имеется команда $q_i a_j \to q_k a_l C$ и при этом,
а) если $C =$ Н, то $w_1' = w_1, w_2' = w_2$ и $a_{j'} = a_l$;
б) если $C =$ Л, то $w_1 = w_1' a$, $a_{j'} = a$, $w_2' = a_l w_2$ (если $w_1 = \varepsilon$, то $w_1' = \varepsilon$ и $a_{j'} = \wedge$);
в) если $C =$ П, то $w_2 = a w_2'$, $a_{j'} = a$, $w_1' = w_1 a_l$ (если $w_2 = \varepsilon$, то $w_2' = \varepsilon$ и $a_{j'} = \wedge$).
$\vdash_{\mathcal{M}}^*$ — это рефлексивное и транзитивное замыкание отношения $\vdash_{\mathcal{M}}$, а $K \vdash_{\mathcal{M}}^n K'$ означает, что конфигурация K за n шагов переходит в K'.
Вычисление м.Т. $\mathcal{M}$ на входе w — это конечная или бесконечная последовательность конфигураций $K_0 \vdash K_1 \vdash \ldots \vdash K_t \vdash K_{t+1} \ldots$ такая, что $K_0 = q_0 w$ — начальная конфигурация.

Эта последовательность конечна, когда её последняя конфигурация $K_n = v_1 q_f v_2$ заключительная. В этом случае вычисление назовем *результативным*, а слово $v = v_1 v_2$ – его *результатом на входе* w.
Время $t_{\mathcal{M}}(w)$ результативного вычисления $\mathcal{M}$ на входе w равно числу шагов (тактов) этого вычисления n, а *память* этого вычисления $s_{\mathcal{M}}(w)$ равна числу различных ячеек ленты, посещённых головкой $\mathcal{M}$ в процессе вычисления. Для оценки сложности "в худшем случае" используются функции $t_{\mathcal{M}}(n) = \max\{t_{\mathcal{M}}(w) \mid |w| = n\}$ и $s_{\mathcal{M}}(n) = \max\{s_{\mathcal{M}}(w) \mid |w| = n\}$.
М.Т. $\mathcal{M}$ *вычисляет частичную словарную функцию* $f: \Sigma^* \to \Sigma^*$, если для каждого слова $w \in \delta_f$ существует результативное вычисление $\mathcal{M}$ с результатом $f(w)$, а если $f(w) = \infty$, то вычисление $\mathcal{M}$ на входе w бесконечно.
Скажем, что две м.Т. $\mathcal{M}_1$ и $\mathcal{M}_2$ *эквивалентны,* если они вычисляют одинаковые функции.
Унарное кодирование: число n будет представляться как слово из n палочек $|^n$ ($n = 0$ кодируется пустым символом $\wedge$), а последовательные аргументы отделяются символами $*$.
Скажем, что м.Т. $\mathcal{M}$ *вычисляет частичную арифметическую функцию* $f: \mathbf{N}^k \to \mathbf{N}$, если для любого набора чисел $(x_1, x_2, \ldots, x_k)$, на котором f определена, существует результативное вычисление $\mathcal{M}$ на входе $|^{x_1} * |^{x_2} * \ldots * |^{x_k}$ с результатом $|^{f(x_1, x_2, \ldots, x_k)}$, а если $f(x_1, x_2, \ldots, x_k) = \infty$, то вычисление $\mathcal{M}$ на соответствующем входе бесконечно.
Тьюрингово программирование.
Стандартная заключительная конфигурация м.Т. — это такая заключительная конфигурация, в которой головка наблюдает первый значащий символ результата в 1-й ячейке (т.е. в той же ячейке, где начиналось входное слово).
Односторонняя машина Тьюринга — это такая м.Т. $\mathcal{M}$, головка которой в процессе вычисления никогда не сдвигается левее начальной ячейки (т.е. всегда находится в ячейках с положительными номерами).
Последовательная композиция.
Пусть м.Т. $\mathcal{M}_1$ вычисляет функцию $f(x)$, а м.Т. $\mathcal{M}_2$ — функцию $g(x)$. Тогда м.Т. $\mathcal{M} = \mathcal{M}_1\ ; \mathcal{M}_2$ вычисляет функцию $h(x) = f(g(x))$.
Параллельная композиция.
Пусть м.Т. $\mathcal{M}_1$ вычисляет функцию $f(x)$, а м.Т. $\mathcal{M}_2$ — функцию $g(x)$, и символ $*$ не входит в алфавит м.Т. $\mathcal{M}_1$. Тогда м.Т. $\mathcal{M} = \mathbf{par}_*(\ \mathcal{M}_1,\ \mathcal{M}_2)$ по любому входу вида $x * y$ выдает результат $f(x) * g(y)$, т.е. вычисляет функцию $H(x * y) = f(x) * g(y)$.
Ветвление (условный оператор).
М. Т. Φ называется *распознающей*, если для некоторого алфавита Σ и каждого входа $x \in \Sigma^*$, на котором Φ останавливается, её результат $\Phi(x) \in \{0, 1\}$.
Пусть Φ — распознающая м.Т., м.Т. $\mathcal{M}_1$ вычисляет $f(x)$, а м.Т. $\mathcal{M}_2$ — $g(x)$. Тогда
м.Т. $\mathcal{M} =$ **if** Φ **then** M_1 **else** M_2 **endif**
вычисляет функцию

$$h(x) = \begin{cases} f(x) & \text{при} \quad \Phi(x) = 0 \\ g(x) & \text{при} \quad \Phi(x) = 1 \end{cases}$$

Повторение (цикл).
Пусть Φ — распознающая м.Т., а м.Т. $\mathcal{M}_1$ вычисляет $f(x)$. Тогда
м.Т. $\mathcal{N} =$ **while** Φ **do** M **enddo**
вычисляет функцию $g(x)$, задаваемую выражением

$$g(x) = \textbf{ while } \Phi(x) = 0 \textbf{ do } \quad x := f(x) \textbf{ enddo}$$

Задачи

22.1. Пусть машина Тьюринга M имеет алфавит ленты $\Sigma = \{\wedge, 0, 1\}$, алфавит состояний $Q = \{q, p, r, !\}$, начальное состояние q, заключительное состояние ! и программу Φ:

$q\ 0 \to q\ 0$ *П*	$p\ 0 \to p\ 1$ *Л*	$r\ 0 \to r\ 0$ *Л*
$q\ 1 \to q\ 1$ *П*	$p\ 1 \to r\ 0$ *Л*	$r\ 1 \to r\ 1$ *Л*
$q \wedge \to p \wedge$ *Л*	$p \wedge \to ! \wedge$ *П*	$r \wedge \to ! \wedge$ *П*

В какую заключительную конфигурацию она перейдёт, начав работу в конфигурации $q1100$? Сколько шагов при этом будет сделано?

22.2. Три машины Тьюринга $M_i = \langle \Sigma, Q, P_i, q, ! \rangle$ $(i = 1, 2, 3)$ имеют общий алфавит ленты $\Sigma = \{\wedge, a, b\}$, алфавит состояний $Q = \{q, p, r, s, t, !\}$, начальное состояние q, заключительное состояние ! и следующие программы:

P_1

$Q \setminus \Sigma$	a	b	$\wedge$
q	$p \wedge$ *П*	$!\ b$ *Н*	$! \wedge$ *Н*
p	$p\ a$ *П*	$r\ b$ *П*	$! \wedge$ *Н*
r	$r\ a$ *П*	$! \wedge$ *Н*	$s\ a$ *Л*
s	$s\ a$ *Л*	$s\ b$ *Л*	$t \wedge$ *П*
t	$q \wedge$ *П*	$! \wedge$ *Н*	$! \wedge$ *Н*

P_2

$Q \setminus \Sigma$	a	b	$\wedge$
q	$p \wedge$ *П*	$!\ b$ *Н*	$! \wedge$ *Н*
p	$p\ a$ *П*	$r\ b$ *П*	$! \wedge$ *Н*
r	$r\ a$ *П*	$! \wedge$ *Н*	$s\ a$ *Л*
s	$s\ a$ *Л*	$t\ b$ *Л*	$q \wedge$ *П*
t	$t\ a$ *П*	$! \wedge$ *Н*	$q \wedge$ *П*

P_3

$Q \setminus \Sigma$	a	b	$\wedge$
q	$p \wedge$ *П*	$!\ b$ *Н*	$! \wedge$ *Н*
p	$r \wedge$ *П*	$r\ b$ *П*	$! \wedge$ *Н*
r	$r\ a$ *П*	$t\ b$ *П*	$! \wedge$ *Н*
s	$s\ a$ *Л*	$s\ b$ *Л*	$q \wedge$ *П*
t	$t\ a$ *П*	$! \wedge$ *Н*	$s\ a$ *Л*

Какие из этих машин переводят любую начальную конфигурацию вида $qa^{2n}b$ в заключительную конфигурацию $!ba^n$ $(n \geq 0)$?

22.3. Требуется построить машину Тьюринга **CHANGE**, которая меняет местами аргументы, точнее переводит любую конфигурацию вида $x*qy$ (x и y — слова в алфавите Σ, не содержащем символов $\wedge, *$ и $\#$, q — начальное состояние) в конфигурацию $y*q'x$ (q' — заключительное состояние). Такая машина Тьюринга используется несколько раз в конструкции параллельной композиции машин Тьюринга.

Пусть $Q = \{q, s, p, r, q'\} \cup \{p^a | a \in \Sigma\}$ — множество состояний **CHANGE**. Какие из следующих программ выполняют требуемую работу, т.е. могут быть использованы в качестве программы для **CHANGE**? (В текстах программ a — это произвольный символ из Σ, а b — это произвольный символ из $\Sigma \cup \{*, \#\}$).

P_1 : $qb \to qb\textit{П}$, $q\wedge \to s\#\textit{Л}$, $sb \to sb\textit{Л}$, $s\wedge \to p\wedge\textit{П}$, $pa \to p^a\wedge\textit{П}$, $p* \to r\wedge\textit{П}, p^ab \to p^ab\textit{П}$, $p^a\wedge \to sa\textit{Л}$, $ra \to ra\textit{П}$, $r\# \to q'*\textit{П}$.

P_2 : $qa \to qa\textit{П}$, $q\wedge \to s*\textit{Л}$, $sb \to sb\textit{Л}$, $s\wedge \to p\wedge\textit{П}$, $pa \to p^a\wedge\textit{П}$, $p* \to r\wedge\textit{П}, p^ab \to p^ab\textit{П}$, $p^a\wedge \to sa\textit{Л}$, $ra \to ra\textit{П}$, $r* \to q'*\textit{П}$.

P_3 : $qa \to qa\textit{П}$, $q\wedge \to s*\textit{Л}$, $sb \to sb\textit{Л}$, $s\wedge \to p\wedge\textit{П}$, $pa \to p^a\wedge\textit{П}$, $p* \to q'*\textit{П}, p^ab \to p^ab\textit{П}$, $p^a\wedge \to sa\textit{Л}$.

22.4. Построить машину Тьюринга, которая переводит любую начальную конфигурацию вида qa^nb^k в заключительную конфигурацию $!\ a^{[(n+k)/2]}b^{[(n+k)/2]}$ (! — заключительное состояние). Оцените время работы построенной м.Т.

22.5. Построить машину Тьюринга, которая переводит любую начальную конфигурацию вида $q0^n1^k$ $(n, k \geq 0)$ в заключительную конфигурацию $!\,0$, если $n > k > 0$, и в заключительную конфигурацию $!\,1$, в противном случае (! — заключительное состояние). Оцените время работы построенной м.Т.

22.6. Построить машину Тьюринга, которая переводит любую начальную конфигурацию вида $q|^n \star |^k \star |^m$ $(n, k, m > 0)$ в заключительную конфигурацию $!\,0$, если $0 < n < k + m$, и в заключительную конфигурацию $!\,1$, в противном случае (! — заключительное состояние). Оцените время работы построенной м.Т.

22.7. Построить машину Тьюринга, которая для любого слова $w \in \{a, b\}^*$ переводит начальную конфигурацию вида qw в заключительную конфигурацию $!\ a^n \star b^m$, где n — число букв a, а m — число букв b в слове w (! — заключительное состояние). Оцените время работы построенной м.Т.

22.8. Слово x называется палиндромом (или симметричным), если оно одинаково читается слева направо и справа налево, т.е. $x = x^{-1}$. Построить машину Тьюринга, определяющую по слову x в алфавите $\{1, 2\}$ является ли оно палиндромом, т. е. вычисляющую функцию:

$$f(x) = \begin{cases} 1, & \text{если слово } x \text{ палиндром,} \\ 2, & \text{в противном случае.} \end{cases}$$

Оцените время работы построенной м.Т.

22.9. Построить машину Тьюринга, сравнивающую два слова $x = x_1 x_2 \dots x_n$ и $y = y_1 y_2 \dots y_m$ в алфавите $\{1, 2, 3\}$ лексикографически: $x \prec y \Leftrightarrow \exists i \leq n[(x_1 = y_1) \ \& \ (x_2 = y_2) \ \& \dots \ (x_{i-1} = y_{i-1}) \ \& \ (x_i < y_i)]$ или для некоторого непустого слова x' выполнено $y = xx'$. Эта машина Тьюринга должна вычислять функцию:

$$f(x, y) = \begin{cases} 1, & \text{если } x \prec y, \\ 2, & \text{если } x = y, \\ 3, & \text{если } y \prec x. \end{cases}$$

22.10. Постройте м.Т. для функции копирования, не увеличивая исходный алфавит Σ.

22.11. Построенная а примере 19.3 для функции копирования м.Т. $\mathcal{M}_2$ на перенос каждого символа копируемого слова тратит $2n + 1$ шаг. Поэтому общее время её работы не меньше $2n^2$. Покажите, что для любой константы $k > 1$ можно построить м.Т., которая копирует слова длины n за время не больше $n^2/k + 2n + 1$.

22.12. Постройте программу м.Т., которая выполняла бы перенос непустого слова слева направо в заданное место ленты, т.е. для любого слова $w \in (\Sigma \backslash \{\wedge\})^*$ и $n > 0$ выполняла преобразование конфигураций: $q_1 w \wedge^n \# \ \vdash^* \ \wedge^n q_2 \# w$.

22.13. Лемма 19.1 говорит о существовании для каждой м.Т. $\mathcal{M}$ эквивалентной м.Т. $\mathcal{M}'$ со стандартными заключительными конфигурациями. Завершите её доказательство, достроив программу $\mathcal{M}'$ на этапах 3 и 4.

22.14. Докажите, что односторонняя м.Т. $\mathcal{M}'$, построенная в лемме 19.2, корректно моделирует исходную м.Т. $\mathcal{M}$.

22.15. Другой по сравнению с конструкцией леммы подход к моделированию двусторонней ленты на односторонней заключается в том, чтобы содержимое правой полуленты $\mathcal{M}$ хранить в чётных ячейках $\mathcal{M}'$, а содержимое левой полуленты — в нечётных, поместив в 1-ю ячейку специальный маркер. Постройте программу, реализующую этот подход (её достоинство — увеличение алфавита ленты всего на 1 символ).

22.16. Пусть м.Т. $\mathcal{M}_1$ вычисляет функцию $f(x)$ за время $t_1(x)$, а м.Т. $\mathcal{M}_2$ — функцию $g(x)$ за время $t_2(x)$. Оцените время работы
а) м.Т. $\mathcal{M} = \mathcal{M}_1; \mathcal{M}_2$, вычисляющей функцию $h(x) = f(g(x))$;
б) м.Т. $\mathcal{N} = \mathbf{par}_*(\mathcal{M}_1, \mathcal{M}_2)$, вычисляющей функцию $H(x*y) = f(x)*g(y)$.

22.17. Пусть машины Тьюринга $\mathbf{M}_i$ $(i = 1, 2, 3)$ построены из следующих простых машин Тьюринга:
$\mathbf{Коп}_a$ — копирует вход после разделительного символа $a : w \Rightarrow waw$;
$\mathbf{Зам}(a, b)$ — заменяет первое слева вхождение символа a на b : $w_1aw_2 \Rightarrow w_1bw_2$ $(a \notin w_1)$;
Сум — складывает два аргумента в унарной системе:$|^x * |^y \Rightarrow |^{x+y}$;
Умн — умножает два аргумента в унарной системе: $|^x * |^y \Rightarrow |^{xy}$;
Пуст — не изменяет аргумент:$w \Rightarrow w$
с помощью операций последовательного и параллельного применения следующим образом:
$\mathbf{M}_1 = \mathbf{Коп}_\#; \mathbf{par}_\#(\mathbf{Коп}_*, \mathbf{Коп}_*); \mathbf{par}_\#(\mathbf{Умн},\ \mathbf{Сум}); \mathbf{Зам}(\#, *); \mathbf{Сум}$.
$\mathbf{M}_2 = \mathbf{Коп}_\#; \mathbf{par}_\#(\mathbf{Коп}_*, \mathbf{Коп}_*); \mathbf{par}_\#(\mathbf{Умн},\ \mathbf{Сум}); \mathbf{par}_\#(\mathbf{Коп}_*; \mathbf{Сум}, \mathbf{Пуст});$ $\mathbf{Зам}(\#, *); \mathbf{Сум}$.
$\mathbf{M}_3 = \mathbf{Коп}_\#; \mathbf{par}_\#(\mathbf{Коп}_*; \mathbf{Умн},\ \mathbf{Пуст}); \mathbf{par}_\#(\mathbf{Коп}_*; \mathbf{Сум}\ ,\ \mathbf{Пуст});$ $\mathbf{Зам}(\#, *); \mathbf{Сум}$.

Какие арифметические функций (при унарном кодировании аргумента и результата) вычисляет каждая из $\mathbf{M}_i$ $(i = 1, 2, 3)$?

22.18. Приведённые ниже машины Тьюринга $\mathbf{M}_i$ $(i = 1, 2, 3, 4)$

$\mathbf{M}_1 = \mathbf{Зам}(\wedge, *); \mathbf{Зам}(\wedge, |); \mathbf{while\ Нуль}_{12}\ \mathbf{do\ par}_*(\mathbf{Выч1},\ \mathbf{Коп}_\#;$ $\mathbf{Зам}(\#, |);\ \mathbf{Выч1})\ \mathbf{enddo}; \mathbf{Выб}_{22}$.

$\mathbf{M}_2 = \mathbf{Зам}(\wedge, *); \mathbf{Зам}(\wedge, |); \mathbf{while\ Нуль}_{12}\ \mathbf{do\ par}_*(\mathbf{Выч1},\ \mathbf{Коп}_\#;$ $\mathbf{par}_\#(\mathbf{Пуст},\ \mathbf{Коп}_\#);\ \mathbf{Зам}(\#, |);\ \mathbf{Зам}(\#, |); \mathbf{Выч1};\ \mathbf{Выч1})\ \mathbf{enddo};\ \mathbf{Выб}_{22}$
.

$$
\begin{aligned}
\mathbf{M}_3 = {} & \textbf{if Нуль}_{11} \textbf{ then Пуст} \\
& \quad \textbf{else Коп}_*;\ \textbf{Зам}(\wedge,*); \textbf{Зам}(\wedge,|); \\
& \quad \textbf{while Нуль}_{13} \textbf{ do par}_*(\textbf{Выч1, Коп}_{\#}, \textbf{Пуст}); \\
& \quad \textbf{par}_{\#}(\textbf{Пуст, Умн}); \textbf{Зам}(\#,*)) \textbf{ enddo}; \textbf{Выб}_{33} \\
& \quad \textbf{endif.}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{M}_4 = {} & \textbf{if Нуль}_{11} \textbf{ then Пуст} \\
& \quad \textbf{else Коп}_*;\ \textbf{Зам}(\wedge,*); \\
& \quad \textbf{while Нуль}_{13} \textbf{ do par}_*(\textbf{Выч1, Коп}_{\#}, \textbf{Пуст}); \\
& \quad \textbf{par}_{\#}(\textbf{Пуст,Сум}); \textbf{Зам}(\#,*)) \textbf{ enddo}; \textbf{Выб}_{33} \\
& \quad \textbf{endif.}
\end{aligned}
$$

построены из простых машин Тьюринга **Коп**$_a$, **Зам(a, b), Сум, Умн** и **Пуст**, описанных в задаче 22.17, и машин
Выб$_{in}$ – выбирает i-й аргумент из n аргументов: $x_1 * \dots * x_i * \dots * x_n \Rightarrow x_i$,
Нуль$_{in}$ – выдаёт 1, если i-й аргумент из n аргументов равен $\wedge$ (нулю), и выдаёт 0, если этот аргумент не равен 0 (имеет вид $|^i,\ i > 0$),
Выч1 – вычитает единицу в унарной системе: $|^j \Rightarrow |^{j-1}$ $(| \Rightarrow \wedge, \wedge \Rightarrow \wedge)$.

Какая из этих машин вычисляет функцию $f(x) = 3^x$ в унарном кодировании, т.е. переводит вход $|^x$ в выход $|^{3^x}$?

22.19. Построить программы машин Тьюринга, вычисляющих следующие функции, и оценить время их работы.
а) Перевод из двоичной системы в унарную: $f^{bu}(n_{(2)}) = |^n$.
б) Сложение и вычитание в двоичной системе: $sum(n * m) = n + m$ и $sub(n * m) = n \dot{-} m$.
в) Умножение в двоичной системе: $mul(n * m) = n * m$. (Реализуйте алгоритм умножения "в столбик".)
г) Возведение в степень: $exp(n * m) = n^m$.
д) Извлечение квадратного корня: $sqr(n) = \lfloor \sqrt{n} \rfloor$.
е) Логарифмирование: $log(n) = \lfloor \log_2 n \rfloor$.
ж) Деление: $div(n * m) = \lfloor \frac{n}{m} \rfloor$ $(m \neq 0)$.
з) Остаток от деления: $rest(n * m) = n \textbf{ mod } m$.
и) Функция выбора аргумента: $I^n_m(|^{x_1} * |^{x_2} * \dots * |^{x_n}) = |^{x_m}$ $(1 \leq m \leq n)$.

22.20. Используя машины Тьюринга из предыдущей задачи, построить программы машин Тьюринга, вычисляющих следующие функции.

а) $f_1(x,y) \;=\; \begin{cases} x^2y^3, & \text{если } \; x < y, \\ \lfloor (x+y)/2 \rfloor & \text{в противном случае.} \end{cases}$

б) $f_2(x,y) \;=\; \begin{cases} \lfloor \sqrt{x}\, \rfloor, & \text{если } x+1 \geq y, \\ 2(x+y) & \text{в противном случае.} \end{cases}$

в) $f_3(x,y) \;=\; \begin{cases} \lfloor \log_2(1+\lfloor \frac{x}{y} \rfloor)\, \rfloor, & \text{если } x > 2y, \\ xy & \text{в противном случае.} \end{cases}$

г) $f_4(x,y) \;=\; \begin{cases} \lfloor \sqrt{x}\, \rfloor, & \text{если } 2x \geq y, \\ (x \;\mathbf{mod}\; y) & \text{в противном случае.} \end{cases}$

22.21. Докажите, что всякую арифметическую функцию $f(x)$, вычислимую на некоторой м. Т. $M = \langle Q, \Sigma, P, q_0, q_f \rangle$, можно также вычислить на м. Т. M', алфавит ленты которой содержит лишь два символа: $\wedge$ и $|$. (Указание: используйте для моделирования одного символа из Σ блок из нескольких подряд идущих ячеек, содержащих его код в алфавите $\{\wedge, |\}$ и замените каждую команду M группой команд, обрабатывающих соответствующий блок ячеек).

22.22. *Многоленточные машины Тьюринга.* k-ленточная машина Тьюринга имеет k лент, по каждой из которых перемещается своя головка. Команда такой м.Т. имеет формат:

$$q(a_1, a_2, \ldots, a_k) \rightarrow q'(b_1, b_2, \ldots, b_k)(C_1, C_2, \ldots, C_k),$$

где q и q' — состояния м.Т., a_i — символ, наблюдаемый головкой на i-й ленте, b_i — символ, записываемый головкой на i-й ленте, C_i — сдвиг головки на i-й ленте. Входные данные записываются на первой ленте, а остальные в начальной конфигурации пусты. Результат также считывается с первой ленты.

Покажите, что любую функцию, вычислимую на 2-х (k-) ленточной машине Тьюринга, можно также вычислить и на обычной машине Тьюринга. На сколько при этом увеличится время вычисления?

22.23. *Многоголовочные машины Тьюринга.* k-головочная машина Тьюринга имеет одну ленту, по которой движутся k головок. Команда такой м.Т. имеет формат:

$$q(a_1, a_2, \ldots, a_k) \rightarrow q'(b_1, b_2, \ldots, b_k)(C_1, C_2, \ldots, C_k),$$

где q и q' — состояния м.Т., a_i — символ, наблюдаемый i-й головкой, b_i —

символ, записываемый i-й головкой, C_i — сдвиг i-й головки. Если несколько головок находятся в одной ячейке и пытаются записать в неё разные символы, то "побеждает" головка с наименьшим номером. В начальной конфигурации все головки находятся в первой ячейке с входными данными.

Покажите, что любую функцию, вычислимую на 2-х (k-)головочной машине Тьюринга, можно также вычислить и на обычной машине Тьюринга. На сколько при этом увеличится время вычисления?

22.24. Покажите, что палиндромы (симметричные слова) (см. задачу 22.8) можно распознать на 2-х ленточной или на 2-х головочной машине Тьюринга за время, пропорциональное их длине.

23. Тезис Тьюринга-Чёрча и неразрешимые проблемы

Теорема. Класс частично рекурсивных функций совпадает с классом функций, вычислимых с помощью структурированных программ, и с классом функций, вычислимых на машинах Тьюринга.

Тезис Тьюринга-Чёрча:
Всякий алгоритм может быть задан в виде соответствующей машины Тьюринга или частично рекурсивного определения, а класс вычислимых функций совпадает с классом частично рекурсивных функций и с классом функций, вычислимых на машинах Тьюринга.

Характеристическая функция c_A^k множества $A \subseteq \mathbf{N}^k$:

$$c_A^k(x_1, \ldots, x_k) = \begin{cases} 1, & \textit{если} \;\; (x_1, \ldots, x_k) \in A, \\ 0 & \textit{в противном случае.} \end{cases}$$

Множество $A \subseteq \mathbf{N}^k$ назовем *разрешимым* (или *рекурсивным*), если его характеристическая функция c_A^k вычислима, т.е. является общерекурсивной функцией, в противном случае, A (и связанная с ним проблема) неразрешимо.
Множество натуральных чисел $A \subseteq \mathbf{N}$ называется *рекурсивно перечислимым*, если оно является множеством значений некоторой общерекурсивной функции $f(x)$, т.е. $A = \{f(x) \mid x = 0, 1, 2, \ldots\}$.

Множество (проблема) $A \subseteq \mathbf{N}$ *сводится* к множеству (проблеме) $B \subseteq \mathbf{N}$, если существует общерекурсивная функция $f : \mathbf{N} \to \mathbf{N}$ такая, что $x \in A$ тогда и только тогда, когда $f(x) \in B$. В этом случае пишем $A \leq_m B$ посредством f. Содержательно, "A сводится к B посредством f" означает, что для выяснения того, входит ли x в A, можно эффективно преобразовать x в такие входные данные $y = f(x)$ проблемы B, что при $y \in B$ имеем $x \in A$, а если $y \notin B$, то и $x \notin A$.

Лемма. Если A сводится к B и проблема A неразрешима, то и проблема B неразрешима.

Пусть $A = \{a_0, a_1, \ldots, a_{m-1}\}$ — алфавит, включающий все символы латинского алфавита, цифры, знак пробела (пусть это будет a_0), знаки ';', '=', '<', ':=', а также знаки-ключевые слова **если**, **то**, **конец**, **пока**, **делай** и **всё**. Тогда каждая структурированная программа Π представляет собой некоторое слово $w_\Pi = a_{i_1} a_{i_2} \ldots a_{i_k}$ $(i_1 > 0)$ в алфавите A . *Номер программы* Π — число $n_\Pi = \sum\limits_{j=1}^{k} i_j m^{k-j}$. По тексту программы Π её номер n_Π определяется однозначно.

Каждому числу $n \in \mathbf{N}$ сопоставим структурированную программу Π_n следующим образом: если $n = n_\Pi$ для некоторой программы Π, то $\Pi_n = \Pi$, иначе, т.е. когда n не является "естественным" номером никакой программы, сопоставим ему в качестве Π_n некоторую никогда не останавливающуюся программу P (например, программу $\Pi_5(1)$: $x_1 := x_1;$ **пока** $x_1 = x_1$ **делай** $x_1 := x_1$ **всё**).

Примеры неразрешимых проблем.

Проблема самоприменимости заключается в проверке для каждой программы Π с входной переменной x и выходной переменной y того, остановится ли Π на собственном номере n_Π, т.е. в вычислении всюду определённой функции

$$F_s(x) = \begin{cases} 1, & \textit{если } \Phi_{\Pi_x, y}(n_{\Pi_x}) < \infty, \\ 0 & \textit{в противном случае.} \end{cases}$$

Проблема останова: по произвольной структурированной программе Π определить, завершится ли вычисление Π на входе 0, т.е. вычислить всюду определённую функцию

$$F_{h0}(n) = \begin{cases} 1, & \textit{если } \Phi_{\Pi_n, y}(0) < \infty, \\ 0 & \textit{в противном случае.} \end{cases}$$

Проблема тотальности: по произвольной структурированной программе Π определить, завершает ли она работу при всех значениях входной переменной, т.е. вычислить всюду определённую функцию

$$F_t(n) = \begin{cases} 1, & \textit{если для всех } a\ \Phi_{\Pi_n, y}(a) < \infty, \\ 0 & \textit{в противном случае.} \end{cases}$$

Проблема эквивалентности: по произвольным двум структурированным программам Π и Π' определить, эквивалентны ли они, т.е. вычисляют ли они одну и ту же функцию:

$$F_{eq}(n,m) = \begin{cases} 1, & \textit{если для всех } a\ \Phi_{\Pi_n,y}(a) = \Phi_{\Pi_m,y}(a), \\ 0 & \textit{в противном случае.} \end{cases}$$

Проблемы оптимизации текста программы. Одна из возможных оптимизаций (текста) программы состоит в удалении из неё операторов присваивания, которые никогда не работают. Определим соответствующую этой оптимизации функцию:

$$F_{opt1}(n,m) = \begin{cases} 1, & \textit{если существует вход } a, \textit{ при работе на} \\ & \textit{котором в программе } \Pi_n \textit{ срабатывает} \\ & m\textit{-й по счету оператор присваивания,} \\ 0 & \textit{в противном случае.} \end{cases}$$

Задачи

23.1. Докажите, что машины Тьюринга $\mathcal{M}_F$ и $\mathcal{M}_f$, определенные в доказательстве теоремы для примитивной рекурсии и минимизации, действительно правильно реализуют указанные операторы.

23.2. В доказательстве теоремы 20.2 для построения м.Т. M_Π, моделирующей работу структурированной программы Π с переменными $x_1, \ldots, x_m$, используется m-этажная лента, на которой работают
а) м.Т. M_0^i , которые реализуют присваивания вида $x_i := 0$, обнуляющие i-й этаж;
б) м.Т. M_{+1}^i, увеличивающие содержимое i-го этажа на 1;
в) м.Т. M^{ij} $(1 \le i, j \le m)$, которые реализуют присваивание $x_i := x_j$, т.е. переписывают содержимое j-го этажа ленты на i-й.

Постройте программы этих машин Тьюринга.

23.3. Докажите утверждение 1, сформулированное в доказательстве теоремы 20.2, используя индукцию по построению программы Π и соответствующей м.Т. M_Π.

23.4. В доказательстве теоремы о том, что функции, вычислимые машинами Тьюринга, являются ч.р.ф., рассмотрен случай, когда м.Т. $\mathcal{M}$ вычисляет функцию от одного аргумента $f(x)$. Покажите, что теорема верна и в общем случае для функций $f(x_1, \ldots, x_n)$ при любом n.

23.5. Согласно тезису Тьюринга-Чёрча язык структурированных программ является универсальным — для любой вычислимой функции в нём имеется вычисляющая её программа. Всякий язык программирования, в котором выразимы все операторы языка структурированных программ, также является универсальным. Некоторые из операторов языка структурированных программ оказываются “лишними” — они выразимы через остальные, т.е. язык сохраняет универсальность и при их удалении. Определите, какие из следующих видов операторов (по отдельности) можно выразить через остальные операторы языка.
а) $x := x + 1$,
б) $x := 0$,
в) $x := y$,
г) **если** $x < y$ **то** P1 **иначе** P2 **конец**,
д) **если** $x = y$ **то** P1 **иначе** P2 **конец**,
е) **пока** $x < y$ **делай** P **всё**,
ж) **пока** $x = y$ **делай** P **всё**.

23.6. Докажите, что отношение алгоритмической сводимости $\leq_m$ является рефлексивным и транзитивным.

23.7. Доказать алгоритмическую неразрешимость следующих проблем.
а) По произвольной программе П определить, является ли вычисляемая ей функция $\Phi_{\Pi,y}(x)$ константой.
б) По произвольной программе П и числам a и b проверить равенство $\Phi_{\Pi,y}(a) = b$.
в) По произвольной программе П определить, является ли множество значений вычисляемой ею функции $\Phi_{\Pi,y}(x)$ бесконечным.
г) По произвольной паре программ П и П$'$ проверить, что для всех x имеет место неравенство $\Phi_{\Pi,y}(x) > \Phi_{\Pi',y}(x)$.

23.8. Докажите, что
а) пересечение двух разрешимых множеств является разрешимым множеством;
б) объединение двух разрешимых множеств является разрешимым множеством;
в) дополнение разрешимого множества является разрешимым множеством.

23.9. Докажите, что для двух разрешимых множеств A и B их “сумма” $A + B = \{x + y \mid x \in A, y \in B\}$ также является разрешимым множеством.

23.10. Пусть A — разрешимое множество, а $g(x)$ и $h(x)$ являются о.р.ф. Докажите, что функция

$$F(x) = \begin{cases} g(x), & \text{если } x \in A, \\ h(x) & \text{в противном случае} \end{cases}$$

также является общерекурсивной.

23.11. Определим для любой пары множеств A и B их "сочленение" $A \oplus B = \{2x \mid x \in A\} \cup \{2x+1 \mid x \in B\}$. Докажите, что для любого множества C, если $A \leq_m C$ и $B \leq_m C$, то $A \oplus B \leq_m C$.

23.12. а) Докажите, что пересечение любого рекурсивно перечислимого множества со множеством простых чисел также является рекурсивно перечислимым множеством.
б) Докажите, что объединение двух рекурсивно перечислимых множеств также является рекурсивно перечислимым множеством.

Ответы, указания, решения

1. Множества, отношения и функции

1.1. $2^{\varnothing} = \{\,\varnothing\,\}$
$2^{\{\,\varnothing\,\}} = \{\,\varnothing, \{\,\varnothing\,\}\,\}$
1.2. D, F и G
1.3. $(A \cup B) \setminus C = \{\{\varnothing\}, \{a, c, d\}, c, \{a\}, \{b\}\}$, $|(A \cup B) \setminus C| = 5$
1.5. $A \times B = \{(0, a), (0, b), (0, c), (1, a), (1, b), (1, c)\}$
$B \times A = \{(a, 0), (a, 1), (b, 0), (b, 1), (c, 0), (c, 1)\}$
1.6. $A \times B = B \times A$ тогда и только тогда, когда $A = B$ или одно из множеств A, B пусто
$[\Leftarrow]$ Очевидно.
$[\Rightarrow]$ Предположим, что $A \neq B$, $A \neq \varnothing$, $B \neq \varnothing$. Тогда существует такой элемент x, что либо $x \in A$, $x \notin B$, либо $x \notin A$, $x \in B$. Предположим, что имеет место первый случай. Пусть $y \in B$. Получаем $(x, y) \in A \times B$, $(x, y) \notin B \times A$. Второй случай рассматривается аналогично.
1.7. а) $A \setminus B = \{\,0, 1\,\}$, $C \cap D = \{\,a, c\,\}$, $F_1 = \{\,(0, a), (0, c), (1, a), (1, c)\,\}$
1.9. б) Докажем, что $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$. Пусть $x \in A \cap (B \cup C)$. По определению пересечения множеств это значит, что $x \in A$ и $x \in B \cup C$. По определению объединения $x \in B$ или $x \in C$. Если $x \in B$, то $x \in A \cap B$, а значит, $x \in (A \cap B) \cup (A \cap C)$. Если же $x \in C$, то $x \in A \cap C$ и снова $x \in (A \cap B) \cup (A \cap C)$.
Докажем, что $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$. Пусть $x \in (A \cap B) \cup (A \cap C)$. По определению объединения множеств это значит, что $x \in A \cap B$ или $x \in A \cap C$. Если $x \in A \cap B$, то $x \in A$ и $x \in B$. Тогда $x \in B \cup C$, и следовательно, $x \in A \cap (B \cup C)$. Если же $x \in A \cap C$, то $x \in A$ и $x \in C$. Значит, $x \in B \cup C$, и следовательно, $x \in A \cap (B \cup C)$.
1.11. а) Докажем включение $A \times (B \cup C) \subseteq (A \times B) \cup (A \times C)$. Пусть $x \in A \times (B \cup C)$. По определению декартова произведения это значит, что $x = (y, z)$, $y \in A$, $z \in B \cup C$. Тогда $z \in B$ или $z \in C$. Если $z \in B$, то $(y, z) \in A \times B$, поэтому $x \in (A \times B) \cup (A \times C)$. Если $z \in C$, то $(y, z) \in A \times C$, значит, и в этом случае $x \in (A \times B) \cup (A \times C)$.
1.12. а) $\delta_R = \mathbf{N} \setminus \{0\}$, $\rho_R = \mathbf{N}$, $R^{-1} = \{(x, y) \mid x, y \in \mathbf{N}$ и x делится на $y\}$, $R \circ R = R$, $R \circ R^{-1} = \{(x, y) \mid x, y \in \mathbf{N} \setminus \{0\}\}$
б) $\delta_R = \{0, 1, \ldots, 10\}$, $\rho_R = \{0, 1, \ldots, 10\}$, $R^{-1} = R$, $R \circ R = R \circ R^{-1} = \{(x, y) \mid 0 \leq x, y \leq 10\}$
в) $\delta_R = \mathbf{N}$, $\rho_R = \{3x + 1 \mid x \in \mathbf{N}\}$, $R^{-1} = \{(x, y) \mid x = 3y + 1\}$,

$R\circ R=\{(x,y)\mid x,y\in\mathbf{N}$ и $y=9x+4\}$, $R\circ R^{-1}=\{(x,x)\mid x\in\mathbf{N}\}$

г) $\delta_R=\{x\in\mathbf{N}$ и $x\leq 10\}$, $\rho_R=\{x^2\mid x\in\mathbf{N}$ и $x\leq 10\}$, $R^{-1}=\{(x^2,x)\mid x\in\mathbf{N}$ и $x\leq 10\}$, $R\circ R=\{(x,x^4)\mid x\in\mathbf{N}$ и $x\leq 10\}$, $R\circ R^{-1}=\{(x,x)\mid x\in\mathbf{N}$ и $x\leq 10\}$

д) $\delta_R=\{a,b,c,d\}$, $\rho_R=\{b,c,d\}$, $R^{-1}=\{(b,a),(c,b),(d,b),(d,c),(b,d)\}$, $R\circ R=\{(a,c),(a,d),(b,b),(b,d),(c,b),(d,c),(d,d)\}$, $R\circ R^{-1}=\{(a,a),(a,d),(b,b),(b,c),(c,b),(c,c),(d,a),(d,d)\}$

1.13. а) R_1 рефлексивно, симметрично, транзитивно.

б) R_2 рефлексивно, антисимметрично, транзитивно.

в) R_3 антисимметрично, транзитивно.

1.14. P и R

1.15. Все отношения рефлексивны.

1.16. Все отношения симметричны.

1.17. (а) и (г)

1.18. (а) и (г)

1.19. $L=\{((x,y),(x,y'))\mid 1\leq x,y,y'\leq 8, y\neq y'\}\cup\{((x,y),(x',y))\mid 1\leq x,x',y\leq 8, x\neq x'\}$

L и K не являются отношениями эквивалентности, так как они не рефлексивны.

$L\circ L=\{((x,y),(x',y'))\mid 1\leq x,y,x',y'\leq 8\}=S\times S$

1.20. а) Эквивалентность. Классы: $\{1,4\}$, $\{2\}$, $\{3\}$, $\{5\}$.

б) Отношение не рефлексивно.

в) Отношение не транзитивно.

г) Эквивалентность. Классы: $\{1,2,4\}$, $\{3,5\}$.

д) Отношение не рефлексивно.

е) Эквивалентность. Классы: $\{1,3,5\}$, $\{2,4\}$.

ж) Отношение не рефлексивно.

1.21. а) Да.

б) Нет.

1.22. а) Рефлексивно, нет иначе, антисимметрично, транзитивно.

1.23. а) R_1 рефлексивно, симметрично.

б) R_2 рефлексивно, транзитивно.

в) R_3 рефлексивно, симметрично, транзитивно.

1.24. а) R_1 является отношением частичного порядка, но не является отношением линейного порядка. б) R_2 является отношением линейного порядка.

1.25. а) $\rho_f=\mathbf{R}$, взаимно-однозначная функция

б) $\rho_f=[1,\infty)$

в) $\rho_f = \mathbf{R}$, взаимно-однозначная функция
г) $\rho_f = (0, \infty)$, 1-1-функция
д) $\rho_f = [0, \infty)$
е) $\rho_f = [-1, 1]$, взаимно-однозначная функция
ж) $\rho_f = [0, 1]$
1.26. а) Неверно.
б) Верно. Нужно проверить, что для любого y существует x такой, что $(f \circ g)(x) = y$, т.е. $f(g(x)) = y$. Так как f — функция "на", то существует z такой, что $f(z) = y$. Так как g — функция "на", то существует x такой, что $g(x) = z$. Тогда $f(g(x)) = y$.
в) Верно.
г) Неверно.
д) Верно.
е) Верно.
1.28. а) Определим вес точки (x, y) как $w(x, y) = |x| + |y|$. Для любого натурального числа k существует лишь конечное число точек веса k. Поэтому все точки с целочисленными координатами можно упорядочить по весу, а точки одного веса — произвольным образом (например, по первой координате, а точки с равной первой координатой — по второй координате). Тогда взаимно однозначное соответствие множества $\mathbf{N}$ и множества точек задаётся функцией $f(i) = (i + 1)$-й элемент в определённом выше упорядочении.
б) и в) Аналогично (а).
д) Любое рациональное число (кроме 0) единственным образом представляется в виде несократимой дроби $\frac{p}{q}$, где p — целое число, q — натуральное число. Ноль будем записывать в виде $\frac{0}{1}$. Определим вес рационального числа как $w(\frac{p}{q}) = |p| + q$. Рациональные числа можно упорядочить по весу.

2. Метод математической индукции

2.2. а) $7^{n+1} - 1 = 7 \cdot (7^n - 1) + 6$
г) $3^{n+1} + 7^{n+1} - 2 = 3(3^n + 7^n - 2) + 4(7^n + 1)$
2.6. Доказательство индукцией по n.
Базис индукции. Если $n = 1$, то есть всего две области. Одну из них нужно покрасить в белый цвет, а другую — в чёрный.
Индукционный шаг. Предположим, что для n прямых требуемая раскрас-

ка найдена. Пусть проведена ещё одна прямая. Если изменить цвета всех областей, находящихся по одну сторону от новой прямой, на противоположные, то получится правильная раскраска. Действительно, если граница двух областей не лежит на новой прямой, то они либо обе не меняли цвет, либо обе поменяли его. В обоих случаях цвета областей останутся разными. Если же граница двух областей лежит на новой прямой, то раньше они образовывали одну область одного цвета. После перекрашивания цвет одной из новых областей изменится, а цвет другой останется прежним, поэтому они будут покрашены в разные цвета.

2.7. $(n+1)$-я прямая имеет n точек пересечения с предыдущими n прямыми. Значит, она проходит через $n+1$ областей и разбивает каждую область на две части. Поэтому добавится $n+1$ областей. Число областей станет равно $(n^2+n+2)/2+n+1 = ((n+1)^2+(n+1)+2)/2$.

2.9. В доказательстве пропущен базис индукции. Неравенство справедливо при всех $n \geq 3$.

2.12. а) Доказательство индукцией по n.
Базис индукции. Один диск можно переложить за одну операцию.
Индукционный шаг. Пусть имеется $n+1$ дисков. Сначала перенесём верхние n дисков на второй стержень за 2^n-1 операций. Затем переложим самый большой диск на третий стержень и перенесём n дисков на третью палочку. Общее число операций равно $(2^n-1)+1+(2^n-1) = 2^{n+1}-1$.
б) Доказательство индукцией по n.
Базис индукции. Для одного диска требуется одна операция.
Индукционный шаг. Пусть имеется $n+1$ дисков. Чтобы переложить все диски на третий стержень, сначала туда нужно положить самый большой, т.е. нижний, диск. Но чтобы эта операция была возможна, все остальные диски должны находиться на втором стержне. По индукционному предположению требуется не менее чем за 2^n-1 операций. После этого ещё одна операция нужна для перенесения большого диска. Наконец, нужно положить n дисков на третий стержень. Для этого нужно не менее 2^n-1 операций. Поэтому общее число операций оказывается не меньше чем $2^{n+1}-1$.
в) Согласно пунктам (а) и (б) нужно $2^{64}-1$ операций, чтобы перенести 64 диска. Предположим, что на перекладывание одного диска тратится одна секунда. Оценим число 2^{64} снизу.
1 сутки $= 24 \cdot 60 \cdot 60$ секунд $= 86400$ секунд
2^{64} секунд $= (2^{10})^6 \cdot 2^4$ секунд $> 10^{18} \cdot 16$ секунд $= \frac{16 \cdot 10^{18}}{86400}$ суток $> 16 \cdot 10^{13}$ суток $> \frac{16 \cdot 10^{13}}{366}$ лет $> \frac{16 \cdot 10^{13}}{400}$ лет $= \frac{10^{13}}{25}$ лет $= 4 \cdot 10^{12}$ лет

Итак, мир просуществует не менее 4 триллионов лет. А согласно современным данным возраст Вселенной составляет около 14 млрд. лет. Поэтому ожидать конца света в ближайшем будущем не стоит.

3. Элементы комбинаторики

3.1. а) $3^9 = 19683$
б) $2^{27} = 134217728$
в) $3^{27} = 7625597484987$
3.2. Выбрать элемент из множества A_i можно $k_i + 1$ способами (имеется k_i элементов, но можно не брать ни один из них). Всего существует $(k_1 + 1)(k_2 + 2) \dots (k_n + 1)$ способов.
3.3. Если q — делитель p, то $q = p_1^{i_1} p_2^{i_2} \dots p_r^{i_r}$, где $i_j \leq k_j$. Согласно предыдущей задаче существует $\prod\limits_{i=1}^{r} (k_i + 1)$ делителей. Число n является квадратом тогда и только тогда, когда все показатели степеней k_i чётны. А это имеет место тогда и только тогда, когда $t(n)$ нечётно.
3.4. C_{201}^2
3.5. а) $C_8^3 = 56$
б) $C_7^4 = 35$
в) $C_7^5 = 21$
г) $C_5^4 = 5$
д) Всего существует $2^8 = 256$ подмножеств множества A. Меньше трёх элементов содержат $C_8^0 + C_8^1 + C_8^2 = 37$ подмножеств. Поэтому хотя бы три элемента содержат $256 - 37 = 219$ подмножеств.
е) Более шести элементов содержат $C_8^7 + C_8^8 = 9$ подмножеств. Поэтому не более шести элементов содержат $256 - 9 = 247$ подмножеств.
3.6. C_{x+y}^y
3.8. а) 560
б) -2500
3.9. Тождества проверяются непосредственным применением формулы для C_n^k.
3.10. Вычислим отношение C_n^{k+1}/C_n^k:

$$\frac{C_n^{k+1}}{C_n^k} = \frac{n!}{(k+1)!(n-k-1)!} : \frac{n!}{k!(n-k)!} = \frac{n-k}{k+1}.$$

Отсюда следует, что $C_n^{k+1} > C_n^k$ тогда и только тогда, когда $n - k > k + 1$, т.е. $k < \frac{n-1}{2}$; $C_n^{k+1} < C_n^k$ тогда и только тогда, когда $k > \frac{n-1}{2}$; $C_n^{k+1} = C_n^k$ тогда и только тогда, когда $k = \frac{n-1}{2}$. При чётном n $\lfloor n/2 \rfloor = \lceil n/2 \rceil$, а при нечётном n $\lfloor n/2 \rfloor = (n-1)/2$, $\lceil n/2 \rceil = (n+1)/2$. Из полученных соотношений следует, что

$$C_n^0 < C_n^1 < \ldots < C_n^{\lfloor n/2 \rfloor} = C_n^{\lceil n/2 \rceil} > C_n^{\lceil n/2 \rceil + 1} > \ldots > C_n^n.$$

3.11. $\frac{11!}{5!2!2!} = 83160$
3.12. Докажем формулу индукцией по k.
Базис индукции. Если $k = 1$, то $n_1 = n$. Выбрать подмножество можно только одним способом, а формула даёт $\frac{n!}{n!} = 1$.
Индукционный шаг. Предположим, что для k формула верна и докажем её для $k+1$. Число способов выбрать первое подмножество равно $C_n^{n_1}$. После этого выбора останется $n - n_1$ элементов, которые нужно разбить на k подмножеств, содержащих $n_2, \ldots, n_{k+1}$ элементов. По индукционному предположению это можно сделать $\frac{(n-n_1)!}{n_2! \ldots n_{k+1}!}$ способами. Поэтому число разбиений исходного множества на $k+1$ подмножеств равно

$$C_n^{n_1} \cdot \frac{(n-n_1)!}{n_2! \ldots n_{k+1}!} = \frac{n!}{n_1!(n-n_1)!} \cdot \frac{(n-n_1)!}{n_2! \ldots n_{k+1}!} = \frac{n!}{n_1! n_2! \ldots n_{k+1}!}.$$

3.13. 6
3.14. Максимальное число туземцев на острове равно 2^{30}.
3.15. а) $8! = 40320$
б) Существует $6! = 720$ способов переставить буквы f, m, p, r, t, x. Подсчитаем число способов вставить буквы a и c так, чтобы между ними были две буквы. Имеется семь позиций для вставки. Первая вставляемая буква может занимать позиции с первой по пятую. С учётом того, что буквы a и c можно вставить в разном порядке, получаем 10 способов вставки. Поэтому число перестановок равно 7200. Аналогично получаем, что число перестановок, в которых между a и c стоят три буквы равно 5760. Значит, общее число перестановок равно 12960.
в) $8! - 12960 = 27360$
г) $4! \cdot 4! = 576$
д) Существует $4! = 24$ перестановки букв a, c, f, r. Будем рассматривать блок из этих символов как целую букву. Тогда существует $5! = 120$ перестановок, если не учитывать порядок букв a, c, f, r. С учётом порядка получается $120 \cdot 24 = 2880$ перестановок.
3.16. Закодируем последовательность результатов партий последовательностью из единиц и двоек. На i-м месте стоит 1, если i-ю партию выиграла

первая команда, и 2, если выиграла вторая команда. Найдём число последовательностей, в которых есть ровно три единицы и не более двух двоек и которые заканчиваются на единицу (это соответствует победе первой команды). В последовательность из трёх единиц можно вставлять двойки в три позиции (последний символ — единица). Имеется одна последовательность, не содержащая двоек, три последовательности, содержащие одну двойку, и $C_3^2 + 3 = 6$ последовательностей, содержащих две двойки. Всего есть 10 последовательностей. Точно так же имеется десять последовательностей, заканчивающихся победой второй команды. Поэтому всего существует 20 вариантов.

3.17. а) Закодируем каждую покупку последовательностью нулей и единиц. Для каждого типа пирожных запишем столько нулей, сколько было куплено пирожных этого типа. Между двумя блоками нулей поставим единицу в качестве разделителя. Например, если покупатель приобрёл два заварных пирожных, два песочных и три бисквитных, то покупка представляется последовательностью 0010011000. Число вариантов покупки равно числу последовательностей из нулей и единиц длины $3 + 7 = 10$, содержащих ровно три единицы. Это число равно $C_{10}^3 = 120$.

б) 20

3.18. 286

3.19. C_{n-k+1}^k

3.20. C_{n-k-1}^k

3.21. а) $2^5 = 32$

б) $2^6 = 64$

в) 2^{m-1}

3.22. Существует C_n^5 пятиэлементных подмножеств и C_n^4 пятиэлементных подмножеств, содержащих 7. По условию имеем $C_n^5 = 4C_n^4$. Решая уравнение, находим $n = 24$.

3.23. Подсчитаем двумя способами количество вариантов выбора k человек из группы, состоящей из n женщин и m мужчин. С одной стороны, это количество равно C_{n+m}^k. С другой стороны, чтобы выбрать k человек, нужно выбрать i женщин и $k - i$ мужчин для некоторого $0 \le i \le k$. Тогда общее количество вариантов равно $\sum\limits_{i=0}^{k} C_n^i C_m^{k-i}$. Получаем, что $C_{n+m}^k = \sum\limits_{i=0}^{k} C_n^i C_m^{k-i}$.

3.25. $A_n^3 C_{n-3}^5$

3.26. 262080

3.27. Существует $N!$ способов поставить N девушек в ряд. Из любого хоровода можно получить N рядов, разбивая хоровод в N местах. Поэтому существует $N!/N = (N-1)!$ хороводов.

3.28. а) 2 и 4

б) 3 и 4

в) 1 и 5

г.1) $\prod\limits_{j=0}^{k-1} C^r_{N-jr}$

г.2) $A_N^{N-kr} \cdot \prod\limits_{j=0}^{k-2} C^r_{(k-j)r}$

3.29. а) $6! \cdot 5! = 86400$

б) $A_5^2 \cdot 8! = 806400$

в) $5! \cdot A_6^5 = 86400$

г) $10! - 144000 = 3484800$

д) $9! \cdot 2 = 725760$

е) $10! - 2 \cdot 9! = 8! \cdot A_9^2 = 2903040$

3.30. Эта задача сводится к задаче **3.19**. Зафиксируем некоторого рыцаря (пусть это будет сэр Ланселот). Он либо пойдёт спасать принцессу, либо останется в замке. Если он пойдёт в поход, то его соседи не пойдут, поэтому из оставшихся $n-3$ рыцарей нужно выбрать $k-1$ человек. Так как оставшиеся рыцари сидят в ряд, то согласно задаче **3.19** выбор можно осуществить C^{k-1}_{n-k-1} способами . Если же сэр Ланселот не пойдёт спасать принцессу, то нужно выбрать k рыцарей из $n-1$, а это можно сделать C^k_{n-k} способами. Поэтому общее количество вариантов равно $C^{k-1}_{n-k-1} + C^k_{n-k} = \frac{n}{n-k} C^k_{n-k}$. В частности при $n = 12$, $k = 5$ получаем 36.

3.31. Используйте индукцию по n.

3.32. Обозначим через A_i множество чисел из первой сотни, делящихся на i (без нуля). Согласно принципу включения-исключения имеем $|A_2 \cup A_3 \cup A_5| = |A_2| + |A_3| + |A_5| - |A_6| - |A_{10}| - |A_{15}| + |A_{30}|$. Легко подсчитать, что $|A_2| = 50$, $|A_3| = 33$, $|A_5| = 20$, $|A_6| = 16$, $|A_{10}| = 10$, $|A_{15}| = 6$, $|A_{30}| = 3$, поэтому $|A_2 \cup A_3 \cup A_5| = 74$. Следовательно ни на одно из чисел 2, 3, 5 не делятся 26 чисел. Аналогично получаем, что среди первой тысячи ни на одно из чисел 2, 3, 5 не делятся 266 чисел.

3.33. Общее число целочисленных решений уравнения $x+y+z = 11$ равно $C^2_{13} = 78$ (см. задачу **3.18**). Пусть A_x – множество решений, в которых $x > 4$, A_y – множество решений, в которых $y > 3$, A_z – множество решений, в которых $z > 7$. Тогда множество “плохих” решений A равно $A_x \cup A_y \cup A_z$. По принципу включения и исключения (задача **3.31**) имеем $|A| = |A_x| +$

$|A_y|+|A_z|-|A_x\cap A_y|-|A_x\cap A_z|-|A_y\cap A_z|+|A_y\cap A_y\cap A_z|$. Оценим слагаемые в правой части. Для этого заметим, что если некоторая переменная в решении равна k, то сумма двух других переменных равна $(11-k)$ и имеется $(12-k)$ пар их возможных значений. Поэтому $|A_x| = 1+2+\ldots+7 = 28, |A_y| = 1+2+\ldots+8 = 36$ и $A_z = 1+\ldots+4 = 10$. Легко проверить, что $|A_x\cap A_y| = 6, |A_x\cap A_z| = 0, |A_y\cap A_z| = 0$ и $|A_y\cap A_y\cap A_z| = 0$. Отсюда получаем, что $|A| = 28+36+10-6 = 68$. Тогда число решений, удовлетворяющих всем условиям, равно $78-68 = 10$.

3.34. *Указание*: используя принцип включения и исключения, определите число слов длины n в алфавите из k букв, в которых отсутствует хотя бы одна из букв.

3.35. 126000 (см. задачу **3.34**)

3.36. Найдём число перестановок, оставляющих на месте хотя бы один элемент. Обозначим множество перестановок, оставляющих на месте элементы $i_1,\ldots,i_k$, через $A_{i_1\ldots i_k}$. Тогда по принципу включения-исключения $|\bigcup\limits_{i=1}^{n} A_i| = \sum_{i=1}^{n}|A_i| - \sum\limits_{1\le i<j\le n}|A_{ij}| + \sum\limits_{1\le i<j<k\le n}|A_{ijk}| - \cdots + (-1)^{n-1}|A_{1,2,\ldots,n}|$. Вычислим $\sum\limits_{1\le i_1<\cdots<i_k\le n}|A_{i_1\ldots i_k}|$. Если зафиксировать k элементов, то оставшиеся $n-k$ элементов можно переставить $(n-k)!$ способами. Существует C_n^k способов выбрать k неподвижных элементов. Поэтому $\sum\limits_{1\le i_1<\cdots<i_k\le n}|A_{i_1\ldots i_k}| = C_n^k(n-k)! = \frac{n!}{k!}$. Тогда $|\bigcup\limits_{i=1}^{n} A_i| = \frac{n!}{1!}-\frac{n!}{2!}+\cdots+(-1)^{n-1}\frac{n!}{n!}$. Следовательно $|\overline{\bigcup\limits_{i=1}^{n} A_i}| = n!-\frac{n!}{1!}+\frac{n!}{2!}-\cdots+(-1)^n\frac{n!}{n!} = n!(1-\frac{1}{1!}+\frac{1}{2!}-\cdots+\frac{(-1)^n}{n!}) \approx \frac{n!}{e}$, где e — основание натуральных логарифмов.

4. Булевы функции и их представления

4.1. Будем рассматривать единичный куб $B^n = \{0,1\}^n$.

а) В k-мерной грани k координат изменяются, а оставшиеся $n-k$ координат фиксированы. Существует C_n^k способов выбрать переменные координаты. После этого существует 2^{n-k} способов зафиксировать значения оставшихся координат. Всего получается $2^{n-k}C_n^k$ k-мерных граней.

б) Для каждой координаты есть три варианта: либо она равна нулю, либо она равна единице, либо она не зафиксирована. Поэтому общее число граней равно 3^n.

4.3. а) $d(\tilde{a},\tilde{b})$ — это число несовпадающих координат точек $\tilde{a}$ и $\tilde{b}$. Поскольку при перемещении по одному ребру изменяется ровно одна координата, то число шагов не может быть меньше k. Существование пути длины k доказывается индукцией по k. При $k = 0$ вершины совпадают и никуда идти не нужно. Если $d(\tilde{a},\tilde{b}) = k+1$, то сначала сделаем шаг из точки $\tilde{a}$ в соседнюю точку $\tilde{c}$. Тогда $d(\tilde{c},\tilde{b}) = k$, и по индукционному предположению существует путь из $\tilde{c}$ в $\tilde{b}$ длины k. Поэтому путь из $\tilde{a}$ в $\tilde{b}$ имеет длину $k+1$.
б) Доказательство индукцией по k.

4.4. а) $\{(0,x_2,x_3,x_4) \mid x_2,x_3,x_4 \in \{0,1\}\}$, $|N^+_{f_1}| = 8$
б) $\{(x_1,x_2,x_3,1) \mid x_1,x_2,x_3 \in \{0,1\}\}$
в) $|N^+_{f_3}| = 11$
г) $\{(0,x_2,x_3,x_4) \mid x_2,x_3,x_4 \in \{0,1\}\} \cup \{(x_1,0,x_3,x_4) \mid x_1,x_3,x_4 \in \{0,1\}\} \cup$ $\{(x_1,x_2,1,1) \mid x_1,x_2 \in \{0,1\}\}$, $|N^+_{f_4}| = 13$

4.5.

а)

X_1	X_2	X_3	f_1
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

б)

X_1	X_2	X_3	f_2
0	0	0	1
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	0

4.6.

а)

X_1	X_2	X_3	$((X_1$	$\to$	$\neg$	$X_3)$	$\vee$	$(X_2$	$+$	$X_3))$
0	0	0	0	1	1	0	**1**	0	0	0
0	0	1	0	1	0	1	**1**	0	1	1
0	1	0	0	1	1	0	**1**	1	1	0
0	1	1	0	1	0	1	**1**	1	0	1
1	0	0	1	1	1	0	**1**	0	0	0
1	0	1	1	0	0	1	**1**	0	1	1
1	1	0	1	1	1	0	**1**	1	1	0
1	1	1	1	0	0	1	**0**	1	0	1

$dep(\Psi_1) = 3$

б)

X_1	X_2	$(\neg$	$(X_1$	$\mid$	$X_2)$	$\sim$	$(\neg$	X_1	$\wedge$	$X_2))$
0	0	0	0	1	0	**1**	1	0	0	0
0	1	1	0	0	1	**1**	1	0	1	1
1	0	1	1	0	0	**0**	0	1	0	0
1	1	1	1	0	1	**0**	0	1	0	1

$dep(\Psi_2) = 3$
в) $dep(\Psi_3) = 4$
г) $dep(\Psi_4) = 3$
д) $dep(\Psi_5) = 3$
4.7. (д)
4.8. а) 2
4.9. Функция полностью определяется своими значениями на первой половине таблицы. Поэтому число функций, принимающих на противоположных наборах разные значения, равно $2^{2^{n-1}}$.
4.10. $2^{2^n - 2^{n/2} + 1}$
4.11. а) и б) Можно.
в) Нельзя. *Указание*: проверьте значения функций на наборе, состоящем из всех единиц.
4.12. Введём переменные a, b и c. Переменная a (соответственно, b и c) равна 1 тогда и только тогда, когда A преступник (соответственно, B и C). Тогда высказывания (а), (б), (в) описываются следующими формулами: $\Phi_1 = b \to \neg c$, $\Phi_2 = a + c$, $\Phi_3 = \neg c \to \neg a$. Знания Ш. Холмса описываются конъюнкцией этих трёх формул: $\Phi = \Phi_1 \wedge \Phi_2 \wedge \Phi_3$. Построим таблицу значений формулы Φ.

a	b	c	$((b$	$\to$	$\neg$	$c)$	$\wedge$	$(a$	$+$	$c))$	$\wedge$	$(\neg$	c	$\to$	$\neg$	$a)$
0	0	0	0	1	1	0	0	0	0	0	**0**	1	0	1	1	0
0	0	1	0	1	0	1	1	0	1	1	**1**	0	1	1	1	0
0	1	0	1	1	1	0	0	0	0	0	**0**	1	0	1	1	0
0	1	1	1	0	0	1	0	0	1	1	**0**	0	1	1	1	0
1	0	0	0	1	1	0	1	1	1	0	**0**	1	0	0	0	1
1	0	1	0	1	0	1	0	1	0	1	**0**	0	1	1	0	1
1	1	0	1	1	1	0	1	1	1	0	**0**	1	0	0	0	1
1	1	1	1	0	0	1	0	1	0	1	**0**	0	1	1	0	1

Формула Φ принимает значение 1 только при $a = 0$, $b = 0$, $c = 1$. Поэтому Ш. Холмс может заключить, что C — единственный преступник.
4.13. B и C — преступники.

4.14. Запись B ошибочна. Утверждать, что ошибочная запись единственна, нельзя.
4.15. Можно заменить условный оператор на $x = 1$.
4.17. A и C
4.18. Может.
4.19. Вылинявшие джинсы-клёши без карманов.
4.20. Нет.
4.21. У злоумышленника был ключ и сообщник, он уехал в экипаже, свидетель не ошибся.

5. Эквивалентность формул

5.3. Используйте индукцию по n.
5.4. в) $(X + Y) \to (X \wedge \neg Y) \underset{(8)}{\equiv} ((X \wedge \neg Y) \vee (\neg X \wedge Y)) \to (X \wedge \neg Y) \underset{(7)}{\equiv}$ $\neg((X\wedge\neg Y)\vee(\neg X\wedge Y))\vee(X\wedge\neg Y) \underset{(5)}{\equiv} (\neg(X\wedge\neg Y)\wedge\neg(\neg X\wedge Y))\vee(X\wedge\neg Y) \underset{(3)}{\equiv}$ $(\neg(X \wedge \neg Y) \vee (X \wedge \neg Y)) \wedge (\neg(\neg X \wedge Y) \vee (X \wedge \neg Y)) \underset{(6)}{\equiv} 1 \wedge (\neg(\neg X \wedge Y)\vee$ $(X \wedge \neg Y)) \underset{(6)}{\equiv} \neg(\neg X \wedge Y) \vee (X \wedge \neg Y) \underset{(5)}{\equiv} (X \vee \neg Y) \vee (X \wedge \neg Y) \underset{(1)}{\equiv} X \vee (\neg Y \vee$ $(X \wedge \neg Y)) \underset{(\text{П}1)}{\equiv} X \vee \neg Y$
$(\neg X\wedge\neg Y)\vee X \underset{(3)}{\equiv} (\neg X\vee X)\wedge(\neg Y\vee X) \underset{(6)}{\equiv} 1\wedge(\neg Y\vee X) \underset{(6)}{\equiv} \neg Y\vee X \underset{(2)}{\equiv} X\vee\neg Y$
5.5. в) $((A \to B) \wedge (B \to C)) \to (A \to C) \underset{(7)}{\equiv} ((\neg A \vee B) \wedge (\neg B \vee C)) \to$ $(\neg A\vee C) \underset{(7)}{\equiv} \neg((\neg A\vee B)\wedge(\neg B\vee C))\vee(\neg A\vee C) \underset{(5)}{\equiv} (\neg(\neg A\vee B)\vee\neg(\neg B\vee C))\vee$ $(\neg A\vee C) \underset{(5)}{\equiv} ((\neg\neg A\wedge\neg B)\vee(\neg\neg B\vee\neg C))\vee(\neg A\vee C) \underset{(4)}{\equiv} ((A\wedge\neg B)\vee(B\vee\neg C))\vee$ $(\neg A \vee C) \underset{(3)}{\equiv} (((A\vee B)\wedge(A\vee\neg C))\wedge((\neg B\vee B)\wedge(\neg B\vee\neg C)))\vee(\neg A\vee C) \underset{(6)}{\equiv}$ $(((A\vee B)\wedge(A\vee\neg C))\wedge(1\wedge(\neg B\vee\neg C)))\vee(\neg A\vee C) \underset{(6)}{\equiv} (((A\vee B)\wedge(A\vee\neg C))\wedge$ $(\neg B\vee\neg C))\vee(\neg A\vee C) \underset{(3)}{\equiv} ((((A\vee B)\vee\neg A)\wedge((A\vee\neg C)\vee\neg A))\wedge((\neg B\vee\neg C)\vee$ $\neg A))\vee C \underset{(1,2,6)}{\equiv} ((1\wedge 1)\wedge((\neg B\vee\neg C)\vee\neg A))\vee C \underset{(6)}{\equiv} ((\neg B\vee\neg C)\vee\neg A)\vee C \underset{(1,2)}{\equiv}$ $(\neg B \vee (\neg C \vee C)) \vee \neg A) \underset{(6)}{\equiv} (\neg B \vee 1) \vee A \underset{(6)}{\equiv} 1$
5.6. а) Пусть $g(X_1, \ldots, X_n) = f^*(X_1, \ldots, X_n)$. Тогда $g^*(X_1, \ldots, X_n) =$ $\neg g(\neg X_1, \ldots, \neg X_n) = \neg\neg f(\neg\neg X_1, \ldots, \neg\neg X_n) =$
$f(X_1, \ldots, X_n)$.
б) $F^*(X_1, \ldots, X_n) = \neg F(\neg X_1, \ldots, \neg X_n) =$

$\neg f(g_1(\neg X_1, \dots, \neg X_n), \dots, g_m(\neg X_1, \dots, \neg X_n)) =$
$\neg f(\neg\neg g_1(\neg X_1, \dots, \neg X_n), \dots, \neg\neg g_m(\neg X_1, \dots, \neg X_n)) =$
$\neg f(\neg g_1^*(X_1, \dots, X_n), \dots, \neg g_m^*(X_1, \dots, X_n)) =$
$f^*(g_1^*(X_1, \dots, X_n), \dots, g_m^*(X_1, \dots, X_n))$
5.7. $(\neg X)^* = \neg X$
$(X \to Y)^* = \neg X \wedge Y$
$(X + Y)^* = \neg(X + Y)$
$(X \mid Y)^* = X \downarrow Y$
$(X \downarrow Y)^* = X \mid Y$
5.8. а) $\neg X \wedge (Y + Z)$
б) $\neg((X \vee Y) + (\neg X \vee Z))$
5.9. б) Нет.
в) $\lceil k/2^{2^n} \rceil$

6. Дизъюнктивные и конъюнктивные нормальные формы

6.1. Доказательство получается непосредственной подстановкой значений переменных в формулы.
6.2. $\mathcal{D}_f = (\neg X \wedge \neg Y \wedge \neg Z) \vee (\neg X \wedge Y \wedge \neg Z) \vee (\neg X \wedge Y \wedge Z) \vee (X \wedge Y \wedge \neg Z) \vee (X \wedge Y \wedge Z)$
$\mathcal{C}_f = (X \vee Y \vee \neg Z) \wedge (\neg X \vee Y \vee Z) \wedge (\neg X \vee Y \vee \neg Z)$
6.3. *Указание*: используйте тождества $\Phi \equiv (\Phi \wedge X) \vee (\Phi \wedge \neg X)$ и $\Phi \equiv (\Phi \vee X) \wedge (\Phi \vee \neg X)$.
6.4. а) $\mathcal{D}_{\Phi_1} = (\neg X \wedge \neg Y \wedge \neg Z) \vee (\neg X \wedge \neg Y \wedge Z) \vee (X \wedge \neg Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge Z) \vee (X \wedge Y \wedge Z)$
$\mathcal{C}_{\Phi_1} = (X \vee \neg Y \vee Z) \wedge (X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee \neg Y \vee Z)$
б) $\mathcal{D}_{\Phi_2} = (\neg X \wedge \neg Y \wedge Z) \vee (X \wedge \neg Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge Z) \vee (X \wedge Y \wedge \neg Z) \vee (X \wedge Y \wedge Z)$
$\mathcal{C}_{\Phi_2} = (X \vee Y \vee Z) \wedge (X \vee \neg Y \vee Z) \wedge (X \vee \neg Y \vee \neg Z)$
6.5. а) $(X \wedge Y \wedge Z) \vee (X \wedge Y \wedge \neg Z) \vee (\neg X \wedge Y \wedge Z) \vee (\neg X \wedge \neg Y \wedge Z) \vee (X \wedge \neg Y \wedge Z)$
б) $(\neg X \wedge Y \wedge Z) \vee (\neg X \wedge Y \wedge \neg Z) \vee (X \wedge Y \wedge Z) \vee (X \wedge \neg Y \wedge Z) \vee (\neg X \wedge \neg Y \wedge Z) \vee (X \wedge Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge \neg Z)$
6.6. а) $(X \vee Y \vee Z) \wedge (X \vee Y \vee \neg Z) \wedge (\neg X \vee Y \vee Z) \wedge (\neg X \vee \neg Y \vee Z) \wedge (X \vee \neg Y \vee \neg Z)$
б) $(X \vee Y \vee Z) \wedge (X \vee Y \vee \neg Z) \wedge (X \vee \neg Y \vee Z) \wedge (X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee Y \vee Z) \wedge (\neg X \vee Y \vee \neg Z) \wedge (\neg X \vee \neg Y \vee Z) \wedge (\neg X \vee \neg Y \vee \neg Z)$
6.7. а) $(X \vee Y) \wedge (X \vee Z) \wedge (Y \vee Z)$

б) $(\neg X \vee Y) \wedge (\neg X \vee \neg Z) \wedge (X \vee Z) \wedge (Y \vee Z)$

6.10. Используйте индукцию по k.

6.11. а) k переменных можно выбрать C_n^k способами, а отрицания при них можно расставить 2^k способами. Поэтому количество неэквивалентных элементарных конъюнкций длины k равно $2^k C_n^k$.

б) 3^n

6.12. (б) и (в)

6.13. Совершенная ДНФ для функции $odd(X_1, \dots, X_n)$ имеет вид $D = \bigvee(x_1^{\sigma_1} \wedge \dots \wedge x_n^{\sigma_n})$, где дизъюнкция берётся по всем наборам $(\sigma_1, \dots, \sigma_n) \in \{0,1\}^n$, содержащим нечётное число единиц. Существует $C_n^1 + C_n^3 + C_n^5 + \dots = 2^{n-1}$ таких наборов. Заметим, что ни одна элементарная конъюнкция, не вошедшая в D, не является допустимой для *odd*. Действительно, если конъюнкция содержит все n переменных, то число единиц в соответствующем наборе чётно и она не допустима. Если же конъюнкция содержит не все переменные, то она не допустима, так как пропущенным переменным можно придавать любые значения и чётность числа единиц в наборе может быть произвольной. Поэтому все конъюнкции, входящие в D, максимальны для *odd*, а значит, D — сокращённая ДНФ. Минимальность D следует теперь из того, что ни одна из входящих в D конъюнкций не может быть опущена.

6.15. а) $X \wedge \neg Y \wedge Z$

б) $\neg Y \vee (X \wedge \neg Z)$

в) $\neg Z \vee (\neg X \wedge \neg Y)$

6.16. а) $\Phi \equiv \Psi \equiv Y \vee (\neg X \wedge \neg Z)$

б) $\Phi \equiv \Psi \equiv X_1 \wedge \neg X_2 \wedge X_3$

в) $\mathcal{D}_\Phi = (X \wedge \neg Y) \vee (U \wedge \neg Y \wedge Z) \vee (\neg X \wedge Y \wedge \neg Z)$

$\mathcal{D}_\Psi = (X \wedge \neg Y) \vee (U \wedge \neg Y \wedge \neg Z) \vee (\neg X \wedge Y \wedge Z) \vee (U \wedge \neg X \wedge \neg Z)$

7. Многочлены Жегалкина

7.1. а) $\Phi \equiv \Psi \equiv X * Y * Z + X + 1$

б) $\Phi \equiv \Psi \equiv X * Y + X$

в) $\Phi \equiv \Psi \equiv X * Y * Z + X * Z + Y * Z + 1$

г) $\Phi \equiv \Psi \equiv X_1 * X_2 * X_3 + X_1 * X_3 + X_1 + 1$

д) $\Phi \equiv \Psi \equiv X_1 * X_2 * X_3 + X_1 * X_3$

7.2. а) $P = X * Y * Z + Y * Z + X + Y$

б) $P = X * Y * Z + X * Y + Y * Z + 1$

в) $P = X + Y + 1$
г) $P = X * Y * Z + X * Y + X + Y + Z$
7.3. а) $X * Y * Z + X * Y + Y * Z + Y + 1$
б) $1 + X_1 + X_4 + X_1 * X_2 + X_1 * X_3 + X_2 * X_3 + X_2 * X_4 + X_1 * X_2 * X_3 + X_1 * X_2 * X_3 * X_4$
в) $X * Y * Z * U + X * Y * Z + Y * Z * U + Y * Z + X$
7.5. а) $X_1 * X_2 * X_3 + X_1 * X_3 + X_1 + X_2$
б) $X_1 * X_2 + X_1 * X_3 + X_3 + 1$
в) $X * Y * Z + X * Z + 1$
7.7. Так как $P(0, \ldots, 0) = 0$, то свободный член многочлена равен 0. Так как $P(1, \ldots, 1) = 0$, то длина многочлена чётна. Поэтому при нечётном k таких многочленов не существует. При чётном k нужно выбрать k слагаемых из $2^n - 1$. Это можно сделать $C^k_{2^n-1}$ способами.
7.8. *Базис индукции.* $n = 1$. При $k = 1$ $P(X_1) = X_1$, при $k = 2$ $P(X_1) = 1$.
Индукционный шаг. Пусть $k \leq 2^{n+1}$.
а) Предположим, что $k \leq 2^n$. По индукционному предположению существует многочлен
$Q(X_1, \ldots, X_n)$ длины не больше n такой, что $|N^+_Q| = k$. Тогда $P(X_1, \ldots, X_{n+1}) = Q(X_1, \ldots, X_n) * X_{n+1}$.
б) Предположим, что $2^n < k \leq 2^{n+1}$. Пусть $l = 2^{n+1} - k$, тогда $l < 2^n$. По индукционному предположению существует многочлен $Q(X_1, \ldots, X_n)$ длины не больше n такой, что $|N^+_Q| = l$. Тогда $P(X_1, \ldots, X_{n+1}) = Q(X_1, \ldots, X_n) * X_{n+1} + 1$.
7.9. Обозначим через $N(n)$ число таких наборов.
Базис индукции: $n = 1$. $N(1) = 1$
Шаг индукции: $P_{n+1} = P_n(X_1, ..., X_n) + X_1 * X_2 * \ldots * X_n * X_{n+1}$.
при $X_{n+1} = 0$ $P_{n+1}(X_1, \ldots, X_n, 0) = P_n(X_1, ..., X_n)$, а при $X_{n+1} = 1$ имеем $P_{n+1}(X_1, \ldots, X_n, 1) = P_n(X_1, ..., X_n) + X_1 * X_2 * \ldots * X_n = P_{n-1}(X_1, ..., X_{n-1})$. Отсюда $N(n + 1) = N(n) + 2N(n - 1) =$
$\frac{1}{3}(2^{n+1} - (-1)^{n+1}) + \frac{2}{3}(2^n - (-1)^n) = \frac{1}{3}(2^{n+2} - (-1)^{n+2})$.

8. Полные системы функций и теорема Поста

8.1. A и D
8.2. B и C
8.3. $|\mathbf{S}_0| = 2^{2^n-1}$
$|\mathbf{S}_1| = 2^{2^n-1}$

$|\mathbf{S}| = 2^{2^{n-1}}$
$|\mathbf{L}| = 2^{n+1}$
8.4. а) $|\mathbf{S}_0 \cap \mathbf{S}_1| = 2^{2^n - 2}$
б) $|\mathbf{S}_0 \cup \mathbf{S}_1| = 3 \cdot 2^{2^n - 2}$
в) $|\mathbf{S} \cap \mathbf{S}_0| = 2^{2^{n-1}-1}$
г) $|\mathbf{L} \cap \mathbf{S}_0| = 2^n$
д) $|\mathbf{L} \cap \mathbf{S}_1| = 2^n$
е) $|\mathbf{S} \cap \mathbf{S}_0 \cap \mathbf{S}_1| = 2^{2^{n-1}-1}$
ж) $|(\mathbf{S} \setminus \mathbf{S}_0) \cap \mathbf{S}_1| = 0$
8.6. $n+1$, так как такая функция определяется наименьшим числом единиц в наборе, на котором функция равна 1.
8.7. Существуют $C_n^{\lfloor n/2 \rfloor}$ наборов длины n, содержащих $\lfloor n/2 \rfloor$ единиц. Любые два таких набора несравнимы, поэтому значения функции на них можно определить произвольным образом. Это можно сделать $2^{C_n^{\lfloor n/2 \rfloor}}$ способами. На наборах с меньшим числом единиц функция будет равна 0, а на наборах с большим числом единиц — 1.
8.8. 2^n
8.9. $n+2$
8.10. Доказательство индукцией по n.
Базис индукции. При $n = 1$ имеется две функции — X_1 и $X_1 + 1$. В обоих случаях $|N_f^+| = 1$.
Индукционный шаг. Пусть $f(X_1, \ldots, X_{n+1}) = \alpha * X_{n+1} + g(X_1, \ldots, X_n)$, где $\alpha \in \{0, 1\}$. При $\alpha = 0$ из любого набора, на котором g равна 1, можно получить два набора, на котором f равна 1, так как переменная X_{n+1} может принимать любое значение. Поэтому $|N_f^+| = 2|N_g^+| = 2^n$. Пусть $\alpha = 1$. Если $(X_1, \ldots, X_n) \in N_g^+$, то $(X_1, \ldots, X_n, 0) \in N_f^+$, а $(X_1, \ldots, X_n, 1) \in N_f^-$. Если же $(X_1, \ldots, X_n) \in N_g^-$, то $(X_1, \ldots, X_n, 0) \in N_f^-$, а $(X_1, \ldots, X_n, 1) \in N_f^+$. По индукционному предположению $|N_g^+| = 2^{n-1}$, поэтому $|N_g^-| = 2^{n-1}$. Получается $|N_f^+| = |N_g^+| + |N_g^-| = 2^n$.
Обратное неверно. Например, функция $f(X_1, X_2, X_3) = X_1 * X_3 + X_2 * X_3 + X_1 + 1$ не является линейной, но $|N_f^+| = 4$.
8.11. $\neg X \equiv X \downarrow X$
$X \vee Y \equiv \neg(X \downarrow Y)$
$X \wedge Y \equiv \neg(\neg X \vee \neg Y)$
8.12. См. таблицу 5.
8.13. Полными системами являются $\{f_1, f_5\}$, $\{f_1, f_2\}$, $\{f_2, f_3, f_5\}$ и все системы, их содержащие.

Таблица 5. Принадлежность функций f_1, f_2, f_3, f_4, f_5 классам $\mathbf{S_0}, \mathbf{S_1}, \mathbf{S}, \mathbf{M}, \mathbf{L}$.

	f_1	f_2	f_3	f_4	f_5
$\mathbf{S_0}$	$+$	$-$	$+$	$+$	$-$
$\mathbf{S_1}$	$-$	$+$	$+$	$+$	$-$
$\mathbf{S}$	$-$	$-$	$+$	$+$	$+$
$\mathbf{M}$	$-$	$-$	$-$	$-$	$-$
$\mathbf{L}$	$-$	$+$	$-$	$+$	$+$

8.14. Система полна.
$0 = h(X, X, X), \qquad 1 = g(X, X, X), \qquad \neg X = g(0, X, 1),$
$X \to Y = \neg g(X, 0, Y).$
8.15. Система не является полной, но она станет полной, если добавить константу 0.
8.16. $1 = X \to X$
$0 = \neg(X \to X)$
$X \vee Y = \neg X \to Y$
$X \wedge Y = \neg(X \to \neg Y)$
$X \sim Y = (Y \to X) \wedge (X \to Y) \equiv \neg(\neg(Y \to X) \to (X \to Y))$
8.17. f и h
8.18. $\{0, \to\}$

9. Хорновские формулы и задача получения продукции

9.1. (a)
9.6. $Cl(X, F) = \{a, b, c, d, e, f, g, h, k\}$
$6, 9, 7, 3, 5, 8$
9.7. $6, 3, 4, 5, 2$
9.9. *Указание*: для каждого полученного продукта запомните номер процесса, с помощью которого он был получен.
9.10. $Cl(X, F) = \{a, c, d, e, f, g, h\}$
$4, 6, 5, 3, 2$

10. Логика предикатов

10.1. (в)
10.2. $\sigma(\varphi_1) = 1$, $\sigma(\varphi_2) = 0$
10.3. а) $Var_{\text{своб}} = \varnothing$, $Var_{\text{связ}} = \{x, z\}$, $\sigma(\varphi_3) = 0$
б) $Var_{\text{своб}} = \{y\}$, $Var_{\text{связ}} = \{x\}$, $\sigma(\varphi_4) = 1$
в) $Var_{\text{своб}} = \{x\}$, $Var_{\text{связ}} = \{u, v\}$, $\sigma(\varphi_5) = 1$
г) $Var_{\text{своб}} = \{z, n\}$, $Var_{\text{связ}} = \{v\}$, $\sigma(\varphi_6) = 1$
10.4. а) $\psi_1 = \forall x(\textit{человек}(x) \to \exists y(\textit{человек}(y) \wedge \textit{живут\_рядом}(x, y) \wedge (\textit{отец}(y) = \textit{лучший\_друг}(x))))$, $\mathcal{A}_1 \not\models \psi_1$
б) $\psi_2 = \forall x(\textit{человек}(x) \to \forall y \forall z((\textit{живут\_рядом}(x, y) \wedge \textit{живут\_рядом}(x, z)) \to (y = z)))$, $\mathcal{A}_1 \not\models \psi_2$
в) $\psi_3 = \forall x(\textit{человек}(x) \to \forall y((\textit{человек}(y) \wedge \textit{живут\_рядом}(x, y) \wedge \neg(\textit{зарплата}(y) \leq \textit{зарплата}(x))) \to \textit{родственники}(x, y))$, $\mathcal{A}_1 \not\models \psi_3$
г) $\psi_4 = \exists x \exists y \exists z(\textit{человек}(x) \wedge \textit{человек}(y) \wedge \textit{человек}(z) \wedge (\textit{отец}(y) = x) \wedge (\textit{отец}(z) = x) \wedge \neg(y = z))$, $\mathcal{A}_1 \models \psi_4$
д) $\psi_5 = \neg\exists x(\textit{человек}(x) \wedge \exists y \exists z \exists t(\textit{человек}(y) \wedge \textit{человек}(z) \wedge \textit{человек}(t) \wedge (\textit{лучший\_друг}(y) = x) \wedge (\textit{лучший\_друг}(z) = x) \wedge (\textit{лучший\_друг}(t) = x) \wedge \neg(y = z) \wedge \neg(y = t) \wedge \neg(z = t)))$, $\mathcal{A}_1 \models \psi_5$
10.5. а) $\neg(20 \leq \textit{зарплата}(x)) \wedge (\textit{зарплата}(x) \leq 35)$
$\sigma(x) \in \{\textit{джон}, \textit{мария}\}$
б) $(\textit{лучший\_друг}(x) = \textit{лучший\_друг}(y))$
$(\sigma(x), \sigma(y)) \in \{(\textit{ольга}, \textit{мария}), (\textit{мария}, \textit{ольга})\}$
в) $(\textit{зарплата}(x) = n) \wedge \forall y(\textit{человек}(y) \to (\textit{зарплата}(y) \leq \textit{зарплата}(x)))$
$(\sigma(x), \sigma(n)) \in \{(\textit{иван}, 40)\}$
г) $\neg(\textit{зарплата}(x) \leq \textit{зарплата}(\textit{отец}(x)))$
$\sigma(x) \in \{\textit{джон}\}$
10.6. а) $\exists z(\neg(z = 0) \wedge (x + z = y))$
б) $\exists y(x = y + y)$
в) $\neg(y = 1) \wedge \forall u \forall v((y = u * v) \to ((u = 1) \vee (v = 1)))$
г) $\forall z((\exists u(x = u * z) \wedge \exists v(y = v * z)) \to (z = 1))$
д) $\exists u(\neg(u = 0) \wedge (z = x + u)) \wedge \exists v(\neg(v = 0) \wedge (y = z + v))$
е) $\forall x \exists y(\exists z(y = x + z) \wedge \forall u \forall v((y = u * v) \to ((u = 1) \vee (v = 1))))$
ж) $\forall x \forall y(x + y = y + x)$, $\forall x \forall y(x * y = y * x)$
10.7. а) $\mathcal{A} \models \varphi$
б) ψ истинна на σ, если x и y — друзья из одной комнаты. $\mathcal{A} \models \psi$
10.8. а) $\neg\forall x(\textit{студент}(x) \to (\textit{изучает}(x, \textit{анализ}) \wedge \textit{изучает}(x, \textit{история})))$

б) $\exists x$(*студент*$(x) \wedge$ (*оценка*$(x,$ *дискретная_математика*$) \leq 2)$ $\wedge\forall y(($*студент*$(y)\wedge$ (*оценка*$(y,$ *дискретная_математика*$) \leq 2)) \to$ $(x = y)))$

в) $\exists x$(*студент*$(x) \wedge \forall y$(*предмет*$(y) \to$ (*оценка*$(x,y) = 5)) \wedge$ $\forall z(($*студент*$(z)\wedge \forall y_1($*предмет*$(y_1) \to$ (*оценка*$(z,y_1) = 5))) \to (x = z)))$

г) $\exists x\exists y[\exists z($*студент*$(z) \wedge$ (*баллы*$(z,$ *дискретная_математика*$) = x))$ $\wedge\forall u\forall v(($*студент*$(u) \wedge$ (*баллы*$(u,$ *дискретная_математика*$) = v)) \to$ $(v \leq x))\wedge \exists z_1($*студент*$(z_1)\wedge$ (*баллы*$(z_1,$ *информатика*$) = y)) \wedge$ $\forall u_1\forall v_1(($*студент*$(u_1) \wedge$ (*баллы*$(u_1,$ *информатика*$) = v_1)) \to v_1 \leq y)\wedge$ $(x > y)]$

д) $\forall x(\forall y($*вегетарианец*$(y) \to \neg$*любит*$(x,y)) \to$ (*человек*$(x) \wedge$ *странный*$(x)))$

е) $\exists x$(*брадобрей*$(x) \wedge \forall y$(*бреет*$(x,y) \sim \neg$*бреет*$(y,y)))$

ж) $\forall x$(*политик*$(x) \to \exists t\forall y$(*человек*$(y) \to$ *может_обманывать*$(x,y,t)))\wedge$
$\forall x$(*политик*$(x) \to \exists y$(*человек*$(y) \wedge \forall t$ *может_обманывать*$(x,y,t)))\wedge$
$\neg\exists x$(*политик*$(x) \to \forall y\forall t$(*человек*$(y) \to$ *может_обманывать*$(x,y,t)))$

10.9. $\exists x_1 \dots \exists x_k(\bigwedge_{1\leq i<j\leq k} \neg x_i = x_j) \wedge \forall x_1 \dots \forall x_{k+1}(\bigvee_{1\leq i<j\leq k+1} x_i = x_j)$

10.12. а) Пусть $\sigma(\neg\forall x\varphi(x)) = 1$. Это значит, что $\sigma(\forall x\varphi(x)) = 0$. По определению это значит, что не для любого состояния σ', отличающегося от σ лишь значением на x, выполняется $\sigma'(\varphi(x)) = 1$. Поэтому существует состояние σ'', отличающееся от σ только значением на x, такое, что $\sigma''(\varphi(x)) = 0$. Тогда $\sigma''(\neg\varphi(x)) = 1$. По определению отсюда следует, что $\sigma(\exists x\neg\varphi(x)) = 1$. Доказательство в другую сторону аналогично.

10.14. Формула $\forall x\exists y(x < y)$ истинна в арифметике, а формула $\exists y\forall x(x < y)$ ложна.

10.15. а) $\exists z\forall x\exists y\forall u\varphi$

б) $\forall x\exists y\exists u\forall v(\varphi(x,y) \to \psi(u,v))$

в) $\exists t\exists y\exists u\exists v(\neg\varphi(t,y,z) \wedge (\neg\psi(x,u) \vee \psi(x,v)))$ $(\equiv \exists t\exists y\neg\varphi(t,y,z))$

г) $\exists x\forall y\exists u\forall v\forall z(\psi(x,y) \wedge \neg\varphi(u) \wedge \neg\psi(v,z))$ $(\equiv 0)$

д) $\exists x\forall y\forall u\forall v\forall z(\psi(x,y) \vee \neg(\varphi(u,v) \vee \psi(z,u)))$

11. Логика предикатов и базы данных

11.1. а) $\sigma_{A=a_1} r = \{(a_1,7,c_1),(a_1,5,c_2)\}$

б) $\pi_B r\times\pi_D s = \{(7,2),(7,8),(7,7),(7,4),(7,5),(5,2),(5,8),(5,7),(5,4),(5,5),$ $(4,2),(4,8),(4,7),(4,4),(4,5)\}$

в) $r \bowtie s = \{(a_1,7,2),(a_1,7,4),(a_1,5,8),(a_1,5,7),(a_1,5,5),(a_2,7,2),(a_2,7,4)\}$

г) $\pi_{A,C} r \bowtie \pi_C r = \{(a_1, c_1), (a_1, c_2), (a_2, c_1), (a_3, c_3)\}$
д) $r \bowtie_{r.C \neq s.c \wedge r.B > s.D} s = \{(a_1, 7, c_1, c_2, 5), (a_1, 5, c_2, c_1, 2), (a_1, 5, c_2, c_1, 4),$ $(a_2, 7, c_1, c_2, 5), (a_3, 4, c_3, c_1, 2)\}$

11.2. (а) и (б)

11.3.
а) **SELECT** ФИО
FROM Сотрудники
WHERE Оклад > 5500
$\varphi_1(x) = \exists x_1 \exists x_2 \exists x_3 \exists x_4 (\text{Сотрудники}(x_1, x, x_2, x_3, x_4) \wedge (x_4 > 5500))$

ФИО
Иванов А. А.
Сидорова М. И.
Горев С. В.

б) **SELECT** Должность, Оклад
FROM Сотрудники
$\varphi_3(x, y) = \exists x_1 \exists x_2 \exists x_3\ \text{Сотрудники}(x_1, x_2, x_3, x, y)$

Должность	Оклад
менеджер	7000
экономист	5000
зав. складом	6000
экономист	5500
зав. отделом	10000

в) **SELECT** Отдел
FROM Сотрудники
WHERE Оклад > 8000
$\varphi_2(x) = \exists x_1 \exists x_2 \exists x_3 \exists x_4 (\text{Сотрудники}(x_1, x_2, x, x_3, x_4) \wedge (x_4 > 8000))$

Отдел
плановый

г) **SELECT** ФИО
FROM Сотрудники, Комнаты
WHERE Номер=НомерСотрудника **AND** Отдел="торговый" **AND**
Оклад ≥ 6000 **AND** Оклад ≤ 6500 **AND** **NOT** Этаж=3
$\varphi_3(x) = \exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists y_1 \exists y_2 \exists y_3 (\text{Сотрудники}(x_1, x, x_2, x_3, x_4) \wedge$ $\text{Комнаты}(y_1, y_2, y_3) \wedge (x_1 = y_1) \wedge (x_4 \geq 6000) \wedge (x_4 \leq 6500) \wedge \neg(y_2 = 3))$

ФИО
Сидорова М. И.

д) **SELECT** НомерКомнаты
FROM Сотрудники, Комнаты
WHERE Номер=НомерСотрудника **AND** (Оклад<5500 **OR** Оклад>7500)
$\varphi_5(x) = \exists x_1 \exists x_2 \exists x_3 \exists x_4 \exists x_5 \exists y_1 \exists y_2$(Сотрудники$(x_1, x_2, x_3, x_4, x_5) \wedge$ Комнаты$(y_1, y_2, x) \wedge (x_1 = y_1) \wedge ((x_5 < 5500) \vee (x_5 > 7500)))$

НомерКомнаты
7
27

11.4. Выполняются Φ_1 и Φ_3.

11.5. а) $\varphi_1 = \forall x \forall y \forall z \forall t(\textit{Комнаты}(x, y, z) \wedge \textit{Комнаты}(x, t, z)) \to (y = t))$
б) $\varphi_2 = \forall x \forall y \forall z \forall u \forall v(\textit{Сотрудники}(x, y, z, u, v) \to \exists s \exists t \textit{Комнаты}(x, s, t))$
в) $\varphi_3 = \forall x \forall y \forall z((\textit{Комнаты}(x, y, z) \wedge (y = 2)) \to ((10 < z) \wedge (z < 20))) \wedge \forall x \forall y \forall z((\textit{Комнаты}(x, y, z) \wedge (y = 3)) \to (20 < z))$
Выполняется только φ_1.

12. Графы: представления, достижимость и связность

12.1. Матрица смежности:

$$A_G = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Матрица инцидентности:

$$B_G = \begin{pmatrix} 1 & 1 & 2 & -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 2 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Списки смежности:
$L_a : a, b, c$
$L_b : a, b$
$L_c : a, d$

$L_d : b$

12.2. $n(n-1)/2$

12.3. (а) и (б)

12.5. По лемме о рукопожатиях (см. задачу 12.4) имеем $5|V| = 2|E|$. При $|E| = 77$ уравнение не имеет целых решений, поэтому ровно 77 дорог быть не может. При $|E| = 80$ получаем $|V| = 32$. На рис. 9 изображён конкретный граф с 32 вершинами, из каждой вершины которого выходит 5 рёбер.

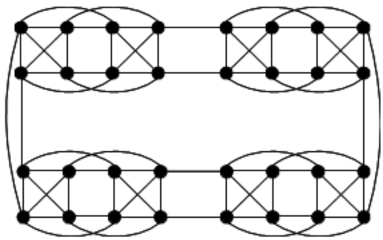


Рис. 9. Граф с 80 рёбрами, из каждой вершины которого выходит 5 рёбер.

12.7. См. рис. 10.

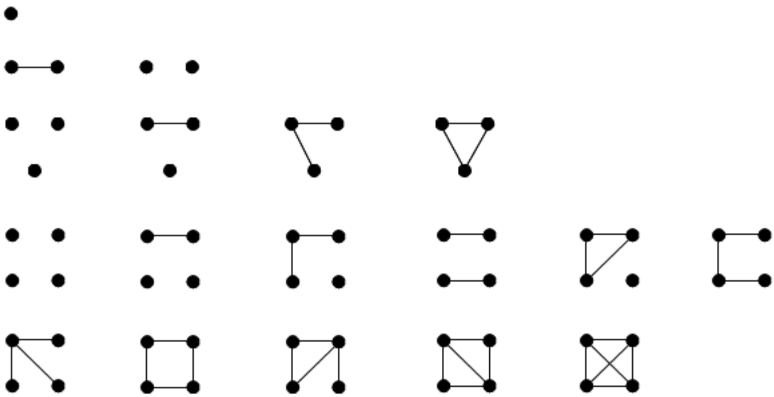


Рис. 10. Неориентированные графы, содержащие не более четырёх вершин.

12.14. *Указание*: докажите, что максимальное число рёбер имеет граф с $k-1$ одновершинными компонентами и одной компонентой — полным $(n-k+1)$-вершинным графом.

12.15. а) Граф с n вершинами и n петлями.
б) Полный n-вершинный граф.

12.16.

$$A_{G^*} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$|E^*| - |E| = 6$

12.17. 3

12.18.

$$A_{G_1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$e_1 = (3,1), e_2 = (3,2), e_3 = (3,4), e_4 = (4,1), e_5 = (4,6), e_6 = (5,4), e_7 = (6,5)$

$$B_{G_1} = \begin{pmatrix} -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Списки смежности.
$L_1 :$
$L_2 :$
$L_3 : 1, 2, 4$
$L_4 : 1, 6$
$L_5 : 4$
$L_6 : 5$

$$A_{G_1^*} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

12.19.

$$A_{G^*} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Базы: $\{a, g\}$, $\{b, g\}$.

12.20. G_1:

$K_1 = \{a, l, n\}$, $K_2 = \{b\}$, $K_3 = \{c\}$, $K_4 = \{d, e, f, g\}$, $K_5 = \{h, m\}$, $K_6 = \{k\}$, $K_7 = \{r\}$

Граф отношения достижимости на компонентах G_1 изображён на рис. 11.

Минимальные компоненты: K_2, K_4, K_7.

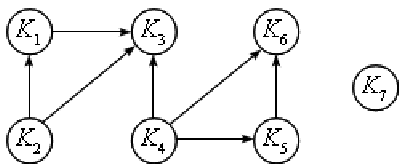


Рис. 11. Граф отношения достижимости на компонентах G_1.

Базы: $\{b, d, r\}$, $\{b, e, r\}$, $\{b, f, r\}$, $\{b, g, r\}$.

G_2:

$K_1 = \{a\}$, $K_2 = \{b\}$, $K_3 = \{c, e, f, g\}$, $K_4 = \{d\}$, $K_5 = \{h, m\}$, $K_6 = \{k\}$, $K_7 = \{l, n\}$

Базы: $\{b, h, l\}$, $\{b, h, n\}$, $\{b, m, l\}$, $\{b, m, n\}$.

13. Деревья

13.3. $V = \{a, b, c\}$, $E = \{(b, c), (c, b)\}$

13.4. Слух узнали $1 + 3 + 3 \cdot 4 + 3 \cdot 4 \cdot 5 = 76$ человек. Было произведено 75 звонков.

13.5. Обозначим число вершин с двумя сыновьями через x, а число вершин с четырьмя сыновьями — через y. По условию $y = 2x$. Подсчитаем общее число вершин двумя способами. Сначала найдём число рёбер в дереве. Из корня выходит 3 ребра, из вершин с двумя сыновьями выходит $2x$ рёбер, а из вершин с четырьмя сыновьями — $4y$ рёбер. Поэтому $|E| = 3+2x+4y = 3+10x$, а значит $|V| = |E|+1 = 4+10x$. С другой стороны $|V| = 1+x+y+38 = 3x+39$. Получаем уравнение $4+10x = 3x+39$, откуда $x = 5$, $y = 10$, $|V| = 54$.

13.6. $|V| = 36$, $|E| = 35$

13.13. 2^h листьев, $2^{h+1} - 1$ вершин

13.14. $\mathit{ПР}(T_1) = abdehicfgkl$

$\mathit{ОБР}(T_1) = dhiebfklgca$

$\mathit{ИНФ}(T_1) = dbheiafckgl$

13.15. $\mathit{ПР}(T_2) = 1\ 6\ 7\ 8\ 2\ 9\ 11\ 10\ 12\ 13\ 14\ 3\ 5\ 4$

$\mathit{ОБР}(T_2) = 7\ 8\ 6\ 11\ 9\ 12\ 13\ 14\ 10\ 5\ 4\ 3\ 2\ 1$

13.17. $x * (z - u/v) + (y - z) * (x + y)$

13.18. См. рис. 12. Удалось сократить 11 вершин.

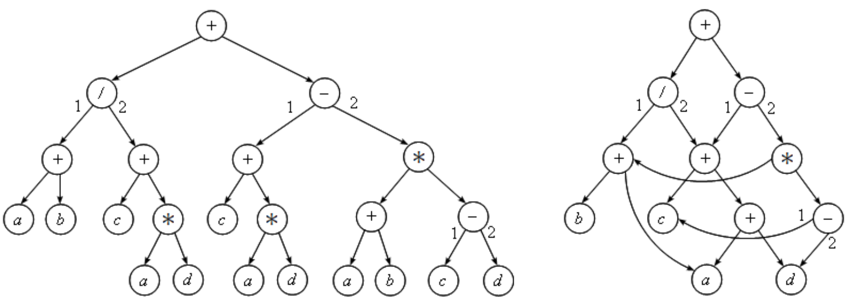


Рис. 12. Дерево и ациклический ориентированный граф для формулы Ф.

14. Алгоритмы на графах

14.2. Нужно удалить ребро (g, m).
Получится Эйлеров цикл $a - c - b - f - e - g - h - a - k - c - m - h - k - f - n - k - b - m - a$.
14.4. Граф не двудольный. Чтобы он стал двудольным, достаточно удалить ребро (b, m).
14.5. На рис. 13 показан граф G, в котором выделены рёбра остовного дерева.

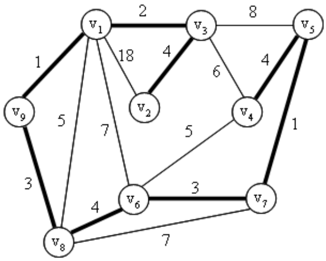


Рис. 13. Минимальное остовное дерево графа G.

14.7. (a, b) и (e, h)
14.8. (b, d), (d, e), (d, f), (f, g), (f, h), (f, i)
14.9. Используйте индукцию по i.
14.14. На рис. 14 показан граф G, в котором выделено глубинное остовное дерево.

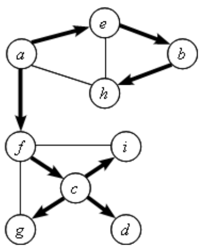


Рис. 14. Глубинное остовное дерево графа G.

v	a	b	c	d	e	f	g	h	i
NUM	1	3	6	7	2	5	8	4	9
ВЕРХ	1	1	5	7	1	5	5	1	5

Первыми будут обнаружены обратное ребро (a, h) и цикл $a - b - h - a$. Мосты: (a, f), (c, d).

14.16. В таблице 6 показана работа алгоритма Дейкстры на графе G. На рис. 15 показан граф G и выделены кратчайшие пути из a в остальные вершины.

Таблица 6. Алгоритм Дейкстры на графе G.

№	S	w	$D[w]$	D					ОТЕЦ				
				b	c	d	e	f	b	c	d	e	f
1	a	c	17	154	17	214	63	∞	a	a	a	a	—
2	a, c	e	50	140	—	209	50	∞	c	a	c	c	—
3	a, c, e	b	140	140	—	209	—	190	c	a	c	c	e
4	a, c, e, b	d	165	—	—	165	—	190	c	a	b	c	e
5	a, c, e, b, d	f	170	—	—	—	—	170	c	a	b	c	d

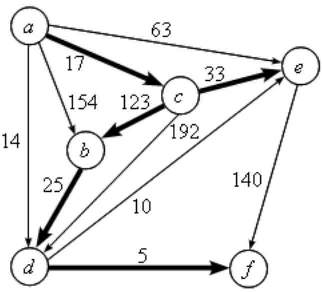


Рис. 15. Кратчайшие пути в графе G.

$c(T) = 203$

14.17. $c(T) = 127$

15. Реализация булевых функций с помощью логических схем

15.1. (0101 1101)

15.2. $\Phi = ((x_1 \wedge x_3) \vee ((x_1 \vee x_2) \wedge x_4)) \wedge \neg(x_3 \vee x_4)$

$v_1 = x_1 \vee x_2;$

$v_2 = x_1 \wedge x_3;$

$v_3 = x_3 \vee x_4;$

$v_4 = v_1 \wedge x_4;$

$v_6 = \neg v_3$;
$v_5 = v_2 \vee v_4$;
$v_7 = v_5 \wedge v_6$.
15.3. $P2$ и $P3$
15.4. $f(x_1, x_2, y_1, y_2) = ((x_1 \wedge y_1) \vee (\neg x_1 \wedge \neg y_1)) \wedge ((x_2 \wedge y_2) \vee (\neg x_2 \wedge \neg y_2))$
См. рис. 16. $D(S_P) = 5$

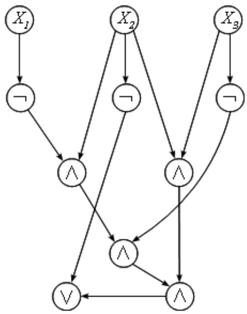


Рис. 16. Логическая схема S_P.

15.7. *Указание:* используйте соотношение $a - b = (2^n - 1) - ((2^n - 1) - a) + b$ и то, что $(2^n - 1) - a = a'$, где биты a' обратны битам a.
15.8. $D(S_+) = 3$, $D(S_{odd}) = 4n - 5$, $D(SUM_1) = 8$, $D(SUM_n) = 9n - 6$
15.9. См. рис. 17.

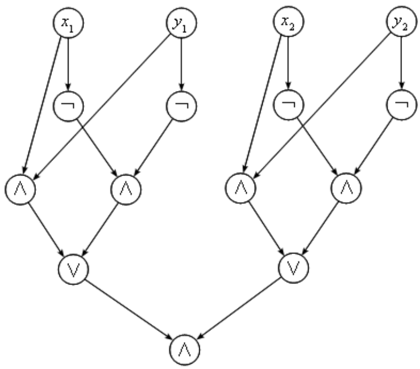


Рис. 17. Логическая схема, задающая выигрыш первого игрока.

15.10. $x_i = 1$ тогда и только тогда, когда i-й член голосует “за”
$y = 1$ тогда и только тогда, когда председатель голосует “за”
$f(x_1, x_2, x_3, y) = (x_1 \wedge x_2 \wedge x_3) \vee (y \wedge (x_1 \vee x_2 \vee x_3))$ См. рис. 18.

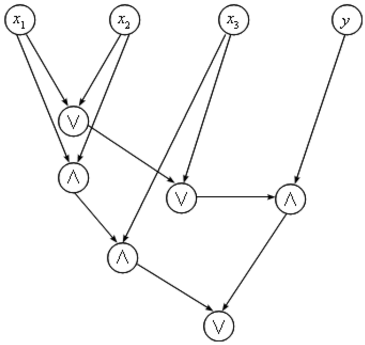


Рис. 18. Логическая схема S_4.

15.11. $f(x_1, x_2, x_3, x_4, x_5) = (x_1 \wedge x_2) \vee ((x_1 \vee x_2) \wedge ((x_3, \wedge (x_4 \vee x_5)) \vee (x_4 \wedge x_5)))$
См. рис. 19.

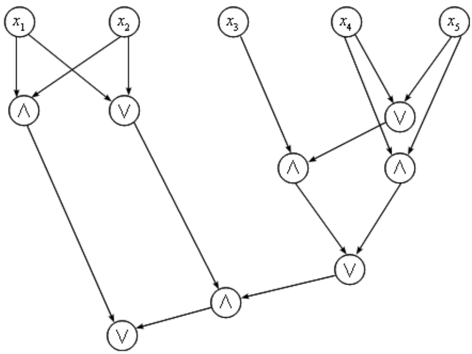


Рис. 19. Логическая схема S_5.

15.12. $c_0 = a_0 \wedge b_0$
$c_1 = \neg(a_0 \wedge b_0 \wedge a_1 \wedge b_1) \wedge ((a_1 \wedge b_0) \vee (a_0 \wedge b_1))$
$c_2 = a_1 \wedge b_1 \wedge \neg(a_0 \wedge b_0)$

$c_3 = a_0 \wedge b_0 \wedge a_1 \wedge b_1$
См. рис. 20.

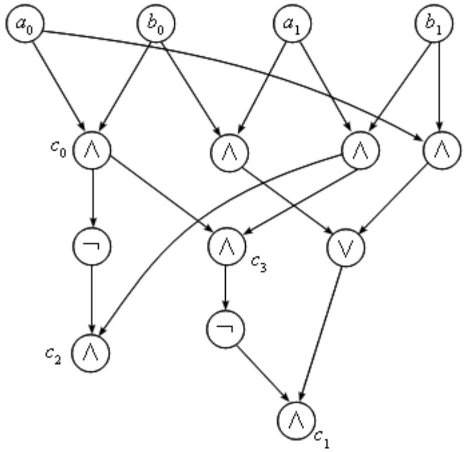


Рис. 20. Логическая схема для умножения двух двухбитовых чисел.

15.13. а) $f_1(x_1, x_2, x_3) = \neg x_1 \wedge x_2 \wedge \neg x_3$ См. рис. 21.

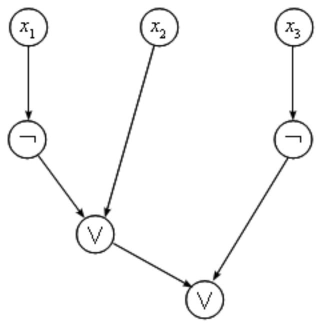


Рис. 21. Логическая схема для функции f_1.

б) $f_2(x_1, x_2, x_3) = \neg(x_2 \vee x_3) \vee (x_2 \wedge x_3)$ См. рис. 22.
в) $f_3(x_1, x_2, x_3) = (\neg x_1 \wedge x_2) \vee (x_1 \wedge \neg(x_2 \vee x_3)) \vee (x_2 \wedge x_3)$ См. рис. 23.

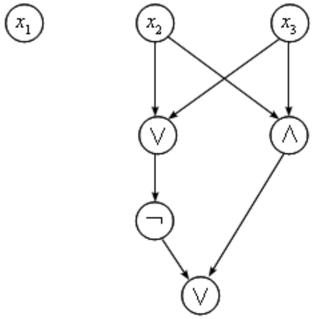


Рис. 22. Логическая схема для функции f_2.

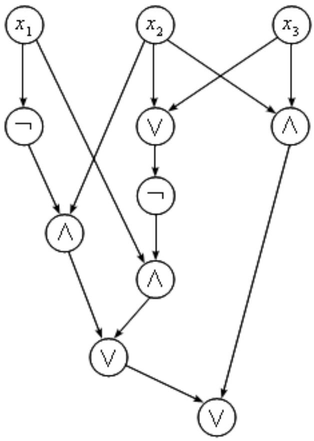


Рис. 23. Логическая схема для функции f_3.

15.14. Используйте следующие формулы.

а) $x \sim y \equiv (x \wedge y) \vee \neg(x \vee y)$

$$f_n(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) = \bigwedge_{i=1}^{n} x_i \sim y_i$$

б) $g_n(x_0, x_1, \ldots, x_{n-1}, y_0, y_1, \ldots, y_{n-1}) = \bigvee_{i=0}^{n-1} ((x_{n-1} \sim y_{n-1}) \wedge \cdots \wedge$

$(x_{i+1} \sim y_{i+1}) \wedge x_i \wedge \neg y_i)$
в) $t_n(x_1, \ldots, x_n) = \bigvee\limits_{1 \le i < j \le n} (x_i \wedge x_j)$
15.15. Используйте следующие формулы.
а) Пусть a_j, b_j — биты числа $\sum\limits_{i=1}^{j} x_i$ **mod** 3. Тогда $a_1 = 0$, $b_1 = x_1$;
$b'_{j+1} = x_j + b_j$, $a'_{j+1} = a_j + (b_j \wedge x_j)$, $a_{j+1} = a'_{j+1} \wedge \neg b'_{j+1}$, $b_{j+1} = b'_{j+1} \wedge \neg a'_{j+1}$.
$f_n(x_1, \ldots, x_n) = a_n \wedge \neg b_n$
б) $a_1 = 0$, $b_1 = x_1$; $a_{j+1} = a_j + (b_j \wedge x_j)$, $b_{j+1} = x_j + b_j$
$f_n(x_1, \ldots, x_n) = a_n \wedge b_n$
в) $f_n(x_1, \ldots, x_n) = odd(x_1, \ldots, x_n)$
15.16. Используйте следующую формулу.
$f(a_{1,1}, a_{1,2}, a_{1,3}, a_{2,1}, a_{2,2}, a_{2,3}, a_{3,1}, a_{3,2}, a_{3,3}) = \bigwedge\limits_{i \ne j, j \ne k, i \ne k} (a_{i,j} \vee (a_{i,k} \wedge a_{k,j}))$

16. Упорядоченные бинарные диаграммы решений (УБДР)

16.1. Сложность УБДР равна $2n+1$.
16.4. См. рис. 24. $f = (1100\ 0110)$

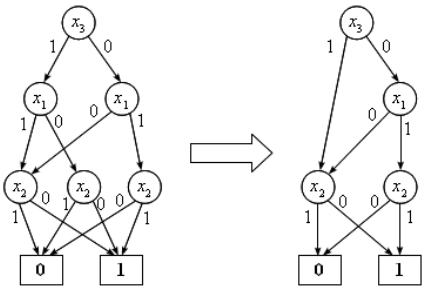


Рис. 24. Сокращение УБДР D.

16.5. а) См. рис. 25.
16.6. а) См. рис. 26.
б) Не зависит.
в) $L(T_n^k) = k(n-k+1)$
16.8. См. рис. 27.

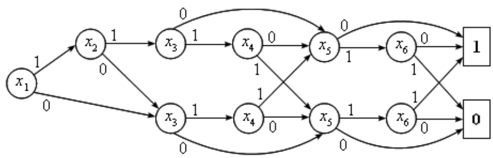


Рис. 25. Минимальная УБДР для функции f.

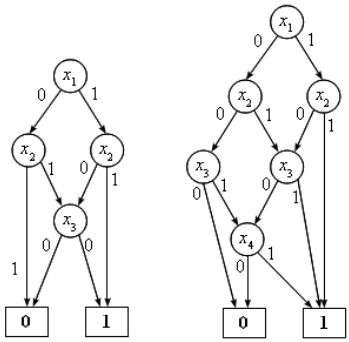


Рис. 26. УБДР для функций T_3^2 и T_4^2.

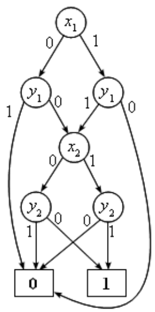


Рис. 27. УБДР для функции $F(x_1, x_2, y_1, y_2)$ из задачи 15.9.

16.9. а)

if $y = 0$ then

  if $z = 0$ then return 0 else return 1 end

else

  if $x = 0$ then return 1 else return 0 end

end

б) При описании ветвящихся программ удобно использовать метки и оператор goto.

1. if $x_1 = 0$ goto 2 else goto 3
2. if $x_2 = 0$ goto 4 else goto 5
3. if $x_2 = 0$ goto 4 else goto 6
4. if $x_3 = 0$ goto 6 else goto 8
5. if $x_4 = 0$ goto 8 else goto 7
6. if $x_4 = 0$ goto 7 else goto 8
7. return 0
8. return 1

16.10. а) Выберем следующий порядок переменных: $x_1 < y_1 < x_2 < y_2 < \cdots < x_n < y_n$. Получающаяся УБДР показана на рис. 28.

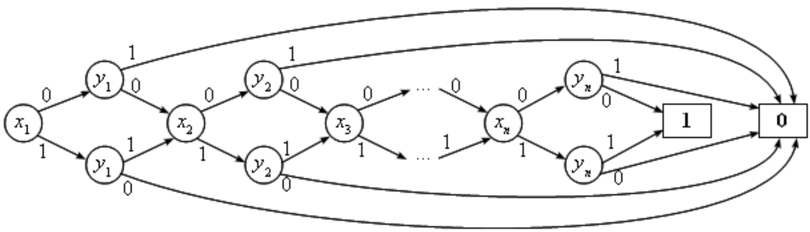


Рис. 28. УБДР для функции f_n из задачи 15.14 (а).

16.11. а) Выберем следующий порядок переменных: $x_1 < x_2 < \cdots < x_n$. Получающаяся УБДР показана на рис. 29.

17. Конечные автоматы: преобразователи и распознаватели

17.1. а) TAPTAP

б) 100001

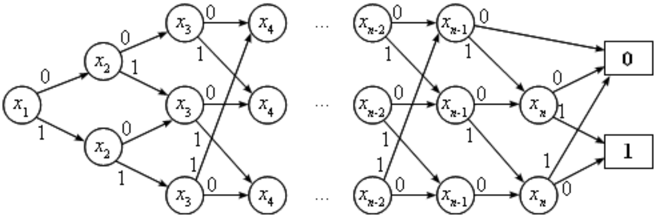


Рис. 29. УБДР для функции f_n из задачи 15.15 (a).

17.2. 3
17.5. V, U
17.6. A_3
17.7. б) См. рис. 30.

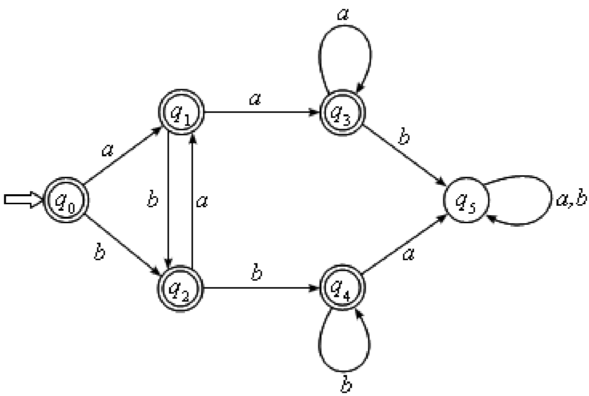
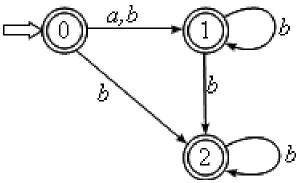


Рис. 30. Конечный автомат для языка L_2.

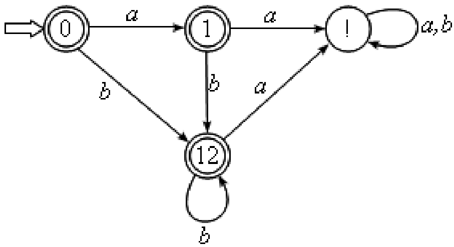
17.9. Используйте индукцию по длине слова w.

17.10. Используя индукцию по длине слов, докажите, что состояниям $\{0,1,2,3\}$ автомата A соответствуют следующие языки:
$L(0) = \{b^i \mid i \geq 0\} \cup \{w \mid w$ заканчивается на $ab^j,\ j \geq 2\}$,
$L(1) = \{b^i a \mid i \geq 0\} \cup \{w \mid w$ заканчивается на $b^2 a\} \cup \{w \mid w$ заканчивается на $a^2\}$,

$L(2) = \{w \mid w$ заканчивается на $ab\}$,
$L(3) = \{w \mid w$ заканчивается на $aba\}$.
17.12. а) w_2 и w_3
17.13. а) Y и R
17.14. а) См. рис. 31.



После удаления ε-переходов



После детерминизации

Рис. 31. Этапы детерминизации автомата M_1.

17.15. M_1 и M_3

18. Регулярные языки и конечные автоматы

18.1. а) $L_1 \cdot L_2 = \{a, aa, ab, aab, aba, abb, abab, abba, abbb, abbab, abbba\}$
б) $L_1 \cdot L_2 = \{a, b, abb, ba, aa, ab, aabb, aba, ababb, abba, abbb, abbabb, abbba\}$
18.2. (в)
18.3. (а) и (е)
18.4. (б), (г) и (д)
18.6. б) Докажем включение $L_{p(qp)^*} \subseteq L_{(pq)^*p}$. Пусть $w \in L_{p(qp)^*}$. Это значит, что $w = uv$ для некоторых слов u и v таких, что $u \in L_p$, $v \in L_{(qp)^*}$. По определению итерации слово v либо пусто, либо имеет вид $v_1v_2\ldots v_k$, где $v_i \in L_{qp}$ для $1 \leq i \leq k$. Если $v = \varepsilon$, то $w = u = \varepsilon u$, а значит, $w \in L_{(pq)^*p}$, так как $\varepsilon \in L_{(pq)^*}$. Пусть v непусто. Так как $v_i \in L_{qp}$, то v_i можно представиить в виде $v_i'v_i''$, где $v_i' \in L_q$, $v_i'' \in L_p$. Итак, $w = uv_1'v_1''v_2'v_2''\ldots v_k'v_k''$. Получаем $uv_1' \in L_{pq}$, $v_i''v_{i+1}' \in L_{pq}$ для $1 \leq i \leq k-1$. Поэтому $uv_1'v_1''\ldots v_{k-1}'v_{k-1}''v_k' \in L_{(pq)^*}$. По определению конкатенации отсюда следует, что $w \in L_{(pq)^*p}$. Включение $L_{(pq)^*p} \subseteq L_{p(qp)^*}$ доказывается аналогично.

18.7. а) Пусть p — регулярное выражение для языка, состоящего из всех слов в алфавите $\{0,1\}$, в которых число нулей чётно и число единиц чётно (см. пример 15.6 из [4]):
$p = (00+11)^*((01+10)(00+11)^*(01+10)(00+11)^*)^*$.
$r_1 = 0p + p0 + (11)^*0p + 1(11)^*0p1$
б) $r_2 = (0+1)^*(001+110)(0+1)^*$
в) $r_3 = (0+1)^*00(0+1)^*$
г) $r_4 = 1^*0^*(1+\varepsilon)$
Нетрудно видеть, что слова языка, задаваемого регулярным выражением r_4, не содержат ни 011, ни 010, т.е. $L_{r_4} \subseteq L_4$. Докажем обратное включение $L_4 \subseteq L_{r_4}$. Заметим, что если слово из языка L_4 содержит подслово 01, то на этом месте слово заканчивается. Пусть $w \in L_4$. Сначала предположим, что w начинается на 0. Тогда сначала в w идут несколько нулей. Если слово на этом заканчивается, то $w = 0^i$ для некоторого i, поэтому $w \in L_{r_4}$. Если же w продолжается, то за блоком нулей может стоять ровно одна единица. В этом случае $w = 0^i1$ для некоторого i и снова $w \in L_{r_4}$. Теперь предположим, что w начинается на 1. Тогда сначала в w стоят несколько единиц. Если на этом w заканчивается, то $w \in L_{r_4}$. Если w не заканчивается, то за блоком единиц следует блок нулей, т.е. w начинается на 1^i0^j для некоторых i и j. Если w не продолжается, то $w = 1^i0^j$, а значит, $w \in L_{r_4}$. Если же w на этом не заканчивается, то далее может стоять ровно одна единица. В этом случае $w = 1^i0^j1$ и $w \in L_{r_4}$. Итак, во всех случаях $w \in L_{r_4}$, а это и означает, что $L_4 \subseteq L_{r_4}$.

18.9. а) 000^*
б) $(0+1)(00)^*$
в) 0^*1
г) 0^*01^*

18.10. Все слова языка L, представляющие правильное сложение, разобьём на два класса: L_0 — слова, задающие сложение двух n-разрядных двоичных чисел, в котором не происходит перенос в $(n+1)$-й разряд, и L_1 — слова, задающие сложение таких n-разрядных двоичных чисел, для которых сумма содержит $(n+1)$ разряд. Пусть слово $w \in L_0$. Тогда его можно представить как конкатенацию подслов $w = w_1^0w_1^1 \dots w_i^0w_i^1 \dots w_k^0w_k^1w_{k+1}^0$, в которой слова w_i^0 $(i = 1,\dots,k,k+1)$ представляют (возможно пустые) участки без переносов, а слова w_i^1 $(i = 1,\dots,k$ — с переносом. Первые задаются регулярным выражением $r_0 = ((0,0,0)+(0,1,1)+(1,0,1))^*$, а вторые — регулярным выражением $r_1 = (1,1,0)((1,0,0)+(0,1,0)+(1,1,0))^*(0,0,1)$ (почему?). Тогда язык L_0 задаётся регулярным выражением $R_0 = (r_0 r_1)^* r_0$.

Постройте выражение R_1, задающее язык L_1, изменив подходящим образом R_0. Тогда язык L будет задаваться выражением $R = R_0 + R_1$.

18.12. $C1$ и $C3$

18.13. $C2$ и $C3$

18.14. а) См. рис. 32.

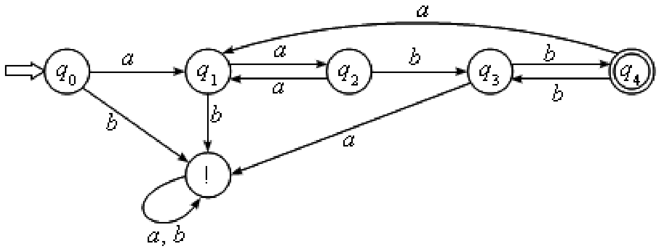


Рис. 32. Конечный автомат, распознающий язык L_{r_1}.

19. Свойства замкнутости класса автоматных языков. Неавтоматные языки

19.1. (в)

19.2. (г)

19.3. w_2

19.4. C_1 и C_2

19.5. См. рис. 33.

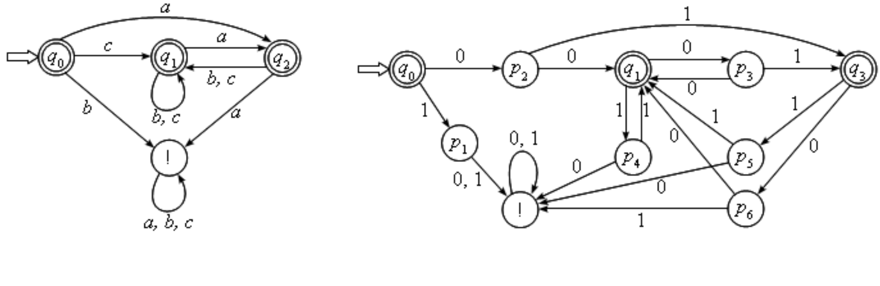


Рис. 33. Конечные автоматы для языков L и $\phi(L)$.

19.8. См. рис. 34.

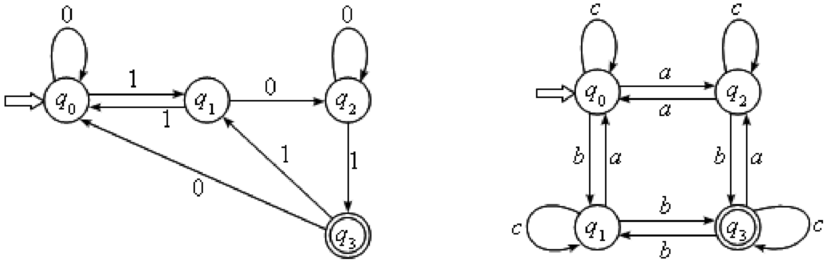


Рис. 34. Конечные автоматы для языков L и $\phi^{-1}(L)$.

19.13. а) Пусть язык L распознаётся детерминированным конечным автоматом $A = \langle \Sigma, Q, q_0, F, \Phi \rangle$. Покажите, что язык ПРЕФ(L) распознаётся автоматом $A' = \langle \Sigma, Q, q_0, F', \Phi \rangle$, где $F' = \{q \mid$ из q в диаграмме автомата A достижимо некоторое состояние $q' \in F\}$.
е) Пусть язык L распознаётся автоматом $A = \langle \Sigma, Q, q_0, F, \Phi \rangle$. Для каждого состояния $q \in Q$ пусть $Q(q) = \{q' \mid$ из q в диаграмме автомата A достижимо q', а из q' достижимо некоторое состояние $q'' \in F\}$. Добавим к A множество пустых переходов $\{q\varepsilon \to q' \mid q' \in Q(q)\}$. Покажите, что полученный таким образом автомат A' распознаёт язык DEL(L).
ж) Обозначим через M автомат для языка L, а через N — автомат для языка CYCLE(L). Состояниями автомата N будут тройки (p, q, i), где pи q — состояния автомата M, $i \in \{0, 1\}$, а также новое начальное состояние. Заключительными состояниями будут $(q, q, 1)$. Из начального состояние по ε-стрелке автомат N переходит в состояния вида $(q, q, 0)$. Первая компонента служит для работы автомата, а во второй он помнит, с какого состояния началась работа. Дойдя до заключительного состояния автомата M, N переходит в начальное состояние M, запоминает в третьей компоненте, что он прошёл через заключительное состояние M, и продолжает работу. Дойдя до конца слова, N проверяет, оказался ли он в состоянии $(q, q, 1)$. Если оказался, то он воспринимает слово, в противном случае отвергает его. Если при чтении слова N не посетил заключительное состояние автомата M, то он также отвергает слово.
19.14. Пусть язык L_1 распознаётся детерминированным конечным автоматом $M = \langle \Sigma, Q, q_0, F, \Phi \rangle$. Покажите, что язык L_1/L_2 распознаётся автоматом $M' = \langle \Sigma, Q, q_0, F', \Phi \rangle$, где $F' = \{q \in Q \mid$ существует слово $x \in L_2$

такое, что$(q, x) \vdash_M^* (q_f, \varepsilon)$для некоторого $q_f \in F\}$.

19.15. Пусть автомат M_1 распознаёт язык L_1, а автомат M_2 распознаёт язык L_2. Состояниями автомата M для языка TAC(L_1, L_2) будут тройки вида (p, q, i), где p — состояние автомата M_1, q — состояние автомата M_2, $i \in \{0, 1\}$. В третьей компоненте автомат помнит, является ли очередная буква чётной или нечётной. На нечётных буквах он работает на первой компоненте состояния как автомат M_1, а на чётных — на второй компоненте как автомат M_2.

19.18. (г)

19.19. L_2, L_5 и L_8

19.21. а) Обозначим язык из задачи через L. Предположим, что язык L автоматный. Тогда для него существует константа n из теоремы о разрастании. Рассмотрим слово $w = a^{n+3}b^n$. Очевидно, что $w \in L$ и $|w| > n$. Поэтому по теореме о разрастании существует разбиение $w = xyz$ такое, что $|xy| \leq n$, $|y| > 0$ и для любого $i \geq 0$ $w_i = xy^iz \in L$. Так как $|xy| \leq n$, то y не содержит символов b. Поэтому $y = a^j$ для некоторого $j > 0$ (так как $|y| > 0$). Рассмотрим слово $w_0 = xz = a^{n+3-j}b^n$. Нетрудно видеть, что $w_0 \notin L$, так как число букв a уменьшилось, а число букв b не изменилось. Это противоречит теореме о разрастании. Значит, предположение неверно, и язык L не является автоматным.

г) Предположим, что язык L автоматный. Тогда для него существует константа k из теоремы о разрастании. Рассмотрим слово $w = a^{2^k}$. Очевидно, что $w \in L$ и $|w| > k$. Поэтому по теореме о разрастании существует разбиение $w = xyz$ такое, что $|xy| \leq n$, $|y| > 0$ и для любого $i \geq 0$ $w_i = xy^iz \in L$. Так как $|y| > 0$, то $y = a^j$ для некоторого $j > 0$. Рассмотрим слово $w_2 = xyyz$. Легко видеть, что $|w_2| > 2^k$. Оценим длину w_2 сверху: $|w_2| = |w| + |y| = 2^k + j \leq 2^k + k < 2^k + 2^k = 2^{k+1}$. Итак, $2^k < |w_2| < 2^{k+1}$, т.е. длина w_2 заключена между соседними степенями двойки. Поэтому $|w_2|$ не является степенью двойки, а значит, $w_2 \notin L$. Это противоречит теореме о разрастании. Значит, предположение неверно, и язык L не является автоматным.

19.23. Рассмотрите умножение чисел вида 2^n.

20. Алгоритмы: структурированные программы

20.1. **P2** и **P5**

20.3. $\mathbf{P}(\sigma)(x) = 6$, $\mathbf{P}(\sigma)(y) = 4$, $\mathbf{P}(\sigma)(z) = 5$

20.4. $\mathbf{P}(\sigma)(x) = 4, \mathbf{P}(\sigma)(y) = 5, \mathbf{P}(\sigma)(z) = 4, \mathbf{P}(\sigma)(u) = 4, \mathbf{P}(\sigma)(v) = 6$

20.5. $\mathbf{P}(\sigma)(x) = 6$, $\mathbf{P}(\sigma)(y) = 5$, $\mathbf{P}(\sigma)(u) = 7$, $\mathbf{P}(\sigma)(v) = 6$

20.6. Π_1 и Π_2

20.7. Π_2 и Π_3

20.8. Π_2

20.9. ж) Приведённая ниже программа Π вычисляет $\lfloor \log_2 x \rfloor$ в переменной y.

$$\begin{array}{l}
\qquad \left. k := 0; y := 0; x' := x; x' := x' + 1; p := 0; p := p + 1; \right\} \Pi' \\
\Pi_1 \left\{ \begin{array}{l}
\textbf{пока } p < x' \textbf{ делай} \\
\quad \left. \begin{array}{l}
u := 0; v := p; \\
\left. \begin{array}{l}
\textbf{пока } u < v \textbf{ делай} \\
\quad u := u + 1; \\
\quad p := p + 1 \\
\textbf{всё}
\end{array} \right\} \Pi_3
\end{array} \right\} \Pi_2 \\
\quad y := k; \\
\quad k := k + 1 \\
\textbf{всё}
\end{array} \right.
\end{array}$$

Докажем правильность программы Π. По определению $\lfloor \log_2 x \rfloor = i$, если $2^i \leq x < 2^{i+1}$. Поэтому $\lfloor \log_2 x \rfloor = \min\{i \mid 2^i > x\} \dot{-} 1$.

Утверждение 1. Для любого состояния σ $\Pi_2(\sigma)(p) = 2\sigma(p)$.

Доказательство. После выполнения первых двух присваиваний в программе Π_2 получится состояние σ', отличающееся от σ только тем, что $\sigma'(u) = 0$, $\sigma'(v) = \sigma(p)$. Пусть $\sigma_0, \sigma_1, \ldots$ — последовательность состояний при работе цикла Π_3 на состоянии σ'. Индукцией по i докажем следующее утверждение: $\sigma_i(u) = i$, $\sigma_i(p) = \sigma(p) + i$, $\sigma_i(t) = \sigma(t)$ для остальных переменных.

Базис индукции. $\sigma_0(u) = \sigma'(u) = 0$, $\sigma_0(p) = \sigma'(p) = \sigma(p)$, $\sigma_0(t) = \sigma'(t) = \sigma(t)$ для любой другой переменной.

Индукционный шаг. По определению семантики цикла получаем $\sigma_{i+1}(u) = \sigma_i(u) + 1 = i + 1$, $\sigma_{i+1}(p) = \sigma_i(p) + 1 = \sigma(p) + i + 1$, а для любой другой пременной $\sigma_{i+1}(t) = \sigma_i(t) = \sigma(t)$.

Пусть $\Pi_2(\sigma') = \sigma_n$. Это значит, что $\sigma_n(u) \geq \sigma_n(v)$, $\sigma_i(u) < \sigma_i(v)$ при $i < n$. Получаем $n \geq \sigma(p)$, $n - 1 < \sigma(p)$, т.е. $n = \sigma(p)$. Тогда $\sigma_n(p) = 2\sigma(p)$, а значит, $\Pi_2(\sigma)(p) = 2\sigma(p)$. $\square$

Утверждение 2. Для любого состояния σ $\Pi(\sigma)(y) = \lfloor \log_2 \sigma(x) \rfloor$.

Доказательство. Пусть $\sigma' = \Pi'(\sigma)$. Тогда $\sigma'(k) = 0$, $\sigma'(y) = 0$, $\sigma'(x') = \sigma(x) + 1$, $\sigma'(p) = 1$. Пусть $\sigma_0, \sigma_1, \ldots$ — последовательность состояний при работе цикла Π_1 на состоянии σ'. Индукцией по i докажем следующее

утверждение: $\sigma_i(p) = 2^i$, $\sigma_i(k) = i$, $\sigma_i(y) = i \dot{-} 1$, $\sigma_i(x') = \sigma(x) + 1$.
Базис индукции. $\sigma_0(p) = \sigma'(p) = 1$, $\sigma_0(k) = \sigma'(k) = 0$, $\sigma_0(y) = \sigma'(y) = 0$, $\sigma_0(x') = \sigma'(x') = \sigma(x) + 1$.
Индукционный шаг. По определению семантики цикла $\sigma_{i+1} = (\Pi_2; y := k; k := k + 1)(\sigma_i)$. Получаем $\sigma_{i+1}(p) = 2\sigma_i(p) = 2 \cdot 2^i = 2^{i+1}$ (по утверждению 1), $\sigma_{i+1}(k) = \sigma_i(k) + 1 = i + 1$, $\sigma_{i+1}(y) = \sigma_i(k) = i = (i + 1) \dot{-} 1$, $\sigma_{i+1}(x') = \sigma_i(x') = \sigma(x) + 1$ (по утверждению 1).
Пусть $\Pi_1(\sigma') = \sigma_n$. Это значит, что $\sigma_n(p) \geq \sigma_n(x')$, $\sigma_i(p) < \sigma_i(x')$ при $i < n$. Получаем $2^n \geq \sigma(x)+1$, $2^{n-1} < \sigma(x)+1$, т.е. $2^n > \sigma(x)$, $2^{n-1} \leq \sigma(x)$. Отсюда следует, что $n - 1 = \lfloor \log_2 \sigma(x) \rfloor$. Но $\sigma_n(y) = n - 1$. Поэтому $\Pi(\sigma)(y) = \lfloor \log_2(x) \rfloor$. □

20.10. а) mC_m^n функций от n переменных, всего $m(2^m - 1)$ функций.

20.13. Программа, используя Π, вычисляет в цикле $f(0)$, $f(1)$, $f(2), \ldots$ до тех пор, пока не обнаружит такой z, что $f(z) = x$.

20.16. *Указание:* докажите следующее утверждение. Пусть Π — программа без циклов, содержащая p присваиваний. Тогда для любого состояния σ и для любой переменной $x \in Var$ существует $y \in Var \cup \{0\}$ такой, что $0 \leq \Pi(\sigma)(x) - \sigma(y) \leq p$ (т.е. всякое выходное значение не очень сильно отличается от некоторого входного значения).
Следствие. Функции $x + y$, $x \dot{-} y$, xy и $\lfloor x/y \rfloor$ не вычисляются никакой структурированной программой без циклов.
Доказательство. Предположим, что некоторая функция вычисляется программой Π в переменной z. Согласно вышеприведённому утверждению существует такое $t \in Var \cup \{0\}$, что $0 \leq \Pi(\sigma)(z) - \sigma(t) \leq p$, где p — число операторов присваивания в Π. Получаем двойное неравенство $\Pi(\sigma)(z) - p \leq \sigma(t) \leq \Pi(\sigma)(z)$.
Рассмотрим сложение. Возьмём состояние σ такое, что $\sigma(x) = 2p$, $\sigma(y) = 2p$. Тогда $\Pi(\sigma)(z) = 4p$. Из неравенства следует, что $\sigma(t) \geq 3p$, что невозможно.
Для остальных функций доказательство аналогично. □

20.15. Будем использовать программы Π_+, $\Pi_{\dot{-}}$ и $\Pi_\times$, вычисляющие в переменной z функции $x + y$, $x \dot{-} y$ и $x * y$, соответственно. Можно считать, что в начале каждой из тих программ рабочие переменные обнуляются явным образом. Кроме того, можно считать, что эти программы не меняют значения входных переменных. Все переменные, используемые ниже, не встречаются в программах Π_+, $\Pi_{\dot{-}}$ и $\Pi_\times$ (кроме x, y, z). Также будем использовать следующую программу Π_0 (при запуске этой программы значение n всегда будет равно 0). Программа Π_0 проверяет условие

“$x > 0$ и $y > 0$”.

если $n < x$ **то**
  если $n < y$ **то**
    $u := 0;\ u := u + 1$
  иначе
    $u := 0$
  конец
иначе
  $u := 0$
конец

Следующая программа вычисляет функцию $A(x, y)$ в переменной z.

$a := 0;\ n := 0;\ e := 0;\ e := e + 1;\ t := e;\ t := t + 1;\ t := t + 1;$
$\Pi_0;$
пока $u = e$ **делай**
  $y' := y;\ y := e;\ \Pi_{\dot{-}};\ x := z;\ y := y';$
  $x' := x;\ y' := y;\ x := y;\ y := e;\ \Pi_{\dot{-}};\ x := x';\ y := z;$
  $\Pi_{\times};$
  $x' := x;\ y' := y;\ x := a;\ y := z;\ \Pi_{+};\ x := x';\ y := y';$
  Π_0
всё;
$b := 1;$
пока $t < x$ **делай**
  $x' := x;\ y' := y;\ x : b;\ y := b;\ \Pi_{+};\ b := z;\ x := x';\ y := y';$
  $y' := y;\ y := e;\ y := y + 1;\ \Pi_{\dot{-}};\ x := z;\ y := y'$
всё;
если $x = z$ **то**
  $x := b;\ y := y + 1;\ \Pi_{\times};\ x := a;\ y := z;\ \Pi_{+}$
иначе
  $x := b;\ y := b;\ \Pi_{+};\ x := z;\ y := z;\ \Pi_{+};\ x := a;\ y := z;\ \Pi_{+}$
конец

В переменных n, e и t хранятся константы 0, 1 и 3. Пока обе переменные x и y положительны, работает первый цикл. В нём используется формула $A(x, y) = A(x - 1, y - 1) + (x - 1)(y - 1)$. В переменной a суммируются слагаемые вида xy до тех пор, пока это возможно. Когда цикл завершит работу, в переменной a окажется “часть” $A(x, y)$, а хотя бы одна из переменных x, y будет равна нулю. Из определения функции $A(x, y)$ следует, что $A(x+2, 0) = 2A(x, 0)$ и $A(1, 0) = 4$. Если после работы первого цикла $x < 2$, то второй цикл не сработает ни разу. Программа сразу попадёт на ветвление и вычислит “оставшуюся часть” $A(x, y)$. Если же $x \geq 2$, то $y = 0$. Тогда второй цикл вычислит множитель перед $A(x \textbf{ mod } 2, 0)$ в переменной b. После этого ветвление завершит вычисление $A(x, y)$.

21. Алгоритмы: частично рекурсивные функции

21.1. $F(x_1, x_2) = 2x_1^2 x_2(4x_2 + 1) + x_2$
21.2. а) $f(x) = x^2$
б) $f(x) = x^2 + x$
в) $f(x) = x^2$
21.3. Используйте индукцию по x.
21.4. а) 2
б) 2
в) 954
21.5. (г)
21.6. (д)
21.7. (г)
21.8. а) $exp = R([s; o], [\times; I_1^3, I_3^3])$
б) $fact = R(1, [\times; [s; I_1^2], I_2^2])$
в) $min = [\dot{-}; I_1^2, \dot{-}]$
г) $max = [+; I_2^2, \dot{-}]$
д) $div = R(o, [+; I_3^3, [\overline{sg}; [rm; I_1^3, [s; I_2^3]]]])$
е) $eq = [\overline{sg}; |-|]$, $\overline{eq} = [sg; |-|]$
ж) $f = [exp; [s; [s; o]], [exp; [s; [s, o]], I_1^1]]$
21.12. $f(x_1, \ldots, x_n) = a * \overline{sg}(x_n - b) + g(x_1, \ldots, x_{n-1}, x_n - b) * sg(x_n - b)$
21.13. а) Первый вариант.
$rt(n, 0) = 0$

$$rt(n, x+1) = \begin{cases} rt(n, x) + 1, & \text{если } x + 1 = (rt(n, x) + 1)^n \\ rt(n, x) & \text{иначе} \end{cases}$$

$rt = [+; I_3^3, [\overline{sg}; [|-|; [s; I_2^3], [exp; [s; I_3^3], I_1^3]]]]$

Второй вариант. $rt(n, x) = \sum\limits_{i=0}^{x} \overline{sg}(i^n \dot{-} x) \dot{-} 1$

б) $log(i, 0) = 0$

$$log(i, x+1) = \begin{cases} log(i, x) + 1, & \text{если } x + 1 = i^{log(i,x)+1} \\ log(i, x) & \text{иначе} \end{cases}$$

$log = [+; I_3^3, [\overline{sg}; [|-|; [s; I_2^3], [exp; I_1^3, [s; I_3^3]]]]]$

в) $t(x) = \sum\limits_{i=0}^{x} \overline{sg}(rm(i, x))$

г) $p(x) = \overline{sg}(|t(x) - 2|)$
д) Пусть p_i — i-е простое число. Справедлива следующая теорема: $p_i \leq 2^{2^i}$. Проведём доказательство индукцией по i. При $i \in \{0,1\}$ $p_0 \leq 2^{2^0}$, $p_1 \leq 2^{2^1}$. Предположим, что $p_0 \leq 2^{2^0}, \dots, p_i \leq 2^{2^i}$, и предположим, что $p_{i+1} > 2^{2^{i+1}}$. Рассмотрим число $q = p_1 \cdot \dots \cdot p_i + 1$. Имеем $q \leq 2^{2^0 + \dots + 2^i} + 1 = 2^{2^{i+1}-1} + 1 < p_{i+1}$. Так как $p_i < q < p_{i+1}$, то q — составное число, но оно не делится ни на одно из чисел $p_1, \dots, p_i$. Противоречие.
$\varphi(n) = \sum\limits_{i=0}^{n} p(i)$ — количество простых чисел на отрезке $[0, n]$
$pn(k) = \mu n \leq 2^{2^k}[|\varphi(n) - k| = 0]$
е) $d(n, m, i) = rm(m, div(m^i, n))$
ж) Пусть $nok(x, y)$ — наименьшее общее кратное чисел x и y. Известно, что $nok(x, y) \leq xy$ и $nod(x, y) nok(x, y) = xy$.
$nok(x, y) = \mu i \leq xy[rm(x, i) + rm(y, i) = 0]$
$nod(x, y) = div(nok(x, y), xy) \cdot sg(x) \cdot sg(y)$ (множители $sg(x)$ и $sg(y)$ добавлены, чтобы выполнялось условие $nod(x, 0) = nod(0, y) = 0$)

21.14. Используйте индукцию по мощности конечного множества, на котором изменяется функция.

21.15. С помощью индукции докажите, что если функция $f(x_1, \dots, x_n)$ получена из функции $o(x)$ и функций выбора с помощью суперпозиции и примитивной рекурсии, то
а) $f(0, \dots, 0) = 0$;
б) $f(x_1, \dots, x_n) \leq \max(x_1, \dots, x_n)$.

21.19. а) $c_2(1, 3) = 13$, $c_2(3, 1) = 23$, $c_3(1, 2, 3) = 3583$, $c_3(0, 4, 1) = 767$
б) $c_2(1, 2) = 9$, $c_2(2, 1) = 11$, $c_3(1, 1, 5) = 351$, $c_3(1, 3, 5) = 90111$
в) $c_n(0, \dots, 0, x) = 2x$, $c_n(1, 0, \dots, 0) = 1$

21.20. а) Номер пары (x, y) — это число предшествующих пар. Паре (x, y) предшествуют пары с меньшей суммой: $(0,0), (0,1), (1,0), \dots, (0, x+y-1), \dots, (x+y-1, 0)$. Количество таких пар равно $1 + 2 + \dots + (x+y) = \frac{(x+y)(x+y+1)}{2}$. Далее идут пары $(0, x+y), (1, x+y-1), \dots, (x-1, y+1)$. Количество таких пар равно x. Поэтому $d(x, y) = \frac{(x+y)(x+y+1)}{2} + x$.
б) Предположим, что пара (x, y) имеет номер z. Тогда $\frac{(x+y)(x+y+1)}{2} + x = z$ $(*)$. Отсюда получаем $2z = (x+y)^2 + 3x + y$, $8z + 1 = (2x + 2y + 1)^2 + 8x = (2x + 2y + 3)^2 - 8y - 8$. Отсюда вытекает, что $2x + 2y + 1 \leq \lfloor\sqrt{8z+1}\rfloor < 2x + 2y + 3$, или $x + y + 1 \leq \frac{\lfloor\sqrt{8z+1}\rfloor + 1}{2} < x + y + 2$. Таким образом,

$$x + y + 1 = \left\lfloor \frac{\lfloor\sqrt{8z+1}\rfloor + 1}{2} \right\rfloor.$$

Сравнивая с формулой (∗), получаем

$$d_1(z) = x = z - \frac{1}{2}\left\lfloor \frac{\lfloor\sqrt{8z+1}\rfloor + 1}{2} \right\rfloor \left\lfloor \frac{\lfloor\sqrt{8z+1}\rfloor - 1}{2} \right\rfloor,$$

$$d_2(z) = y = -z + \frac{1}{2}\left\lfloor \frac{\lfloor\sqrt{8z+1}\rfloor + 5}{2} \right\rfloor \left\lfloor \frac{\lfloor\sqrt{8z+1}\rfloor - 1}{2} \right\rfloor.$$

21.21. а) 3
б) 2

22. Алгоритмы: машины Тьюринга

22.1. !1011, 10 шагов
22.2. M_1 и M_3
22.3. P_1 и P_2
22.10. Очередной символ нужно стирать, переносить в копию и восстанавливать при возврате.
22.12. За один проход нужно запоминать и переносить в копию k подряд идущих символов слова.
22.16. а) $O(t_2(x) + t_1(|g(x)|))$
б) $O(t_2(y) + |x|(|x| + |g(y)|) + t_1(x) + |g(y)|(|f(x)| + |g(y)|))$
22.17. $\mathbf{M}_1 \colon f_1(x) = x^2 + 2x$
$\mathbf{M}_2 \colon f_2(x) = 2x^2 + 2x$
$\mathbf{M}_3 \colon f_1(x) = 2x^2 + x$
22.18. M_2
22.19. д) Будем использовать следующие машины Тьюринга (после каждой машины указано время работы).
$M_0 \colon |^x \Rightarrow |^x * \wedge *,\ t = O(x)$
$M_{+1} \colon |^x * |^y * |^z \Rightarrow |^x * |^{y+1} * |^{z+1},\ t = O(x+y+z)$
$M_{\text{коп}} \colon |^x * |^y * |^z \Rightarrow |^x * |^y * |^{y+z},\ t = O(x + y(y+z))$
$M_f \colon |^x * |^y * |^z \Rightarrow |^{y-1},\ t = O(x+y+z)$
$\Phi \colon |^x * |^y * |^z \Rightarrow \begin{cases} 0, & \text{если } z \le x \\ 1, & \text{иначе} \end{cases},\ t = O((x+y+z)\min(x,z))$

Следующая машина Тьюринга M вычисляет квадратный корень.
M_0;
while Φ **do** $M_{\text{коп}}; M_{\text{коп}}; M_{+1}$ **enddo**;
M_f

Докажем правильность машины Тьюринга M. По определению $\lfloor\sqrt{n}\rfloor = i$, если $i^2 \le n < (i+1)^2$. Поэтому $\lfloor\sqrt{n}\rfloor = \min\{i \mid i^2 > n\} - 1$. Пусть M начинает работу в конфигурации $q_0|^n$. После срабатывания M_0 на ленте будет написано $|^n{*}\wedge{*}$. Индукцией по i докажем следующее утверждение: после i итераций цикла на ленте написано $|^n * |^i * |^{i^2}$, при этом $n \le (i-1)^2$.
Базис индукции. Конфигурация для $i = 0$ приведена выше (считаем, что $|^0 = \wedge$).
Индукционный шаг. Пусть после i итераций цикла на ленте оказалось $|^n{*}|^i{*}|^{i^2}$, $n \le (i-1)^2$. Чтобы сработала $(i+1)$-я итерация, должно выполняться условие $i^2 \le n$. После первого срабатывания м.Т. $M_{\text{коп}}$ на ленте будет написано $|^n * |^i * |^{i^2+i}$, после второго срабатывания — $|^n * |^i * |^{i^2+2i}$. После срабатывания м.Т. M_{+1} на ленте будет написано $|^n * |^{i+1} * |^{i^2+2i+1}$, т.е. $|^n * |^{i+1} * |^{(i+1)^2}$.
Цикл остановится, когда впервые нарушится условие $i^2 \le n$, т.е. окажется, что $i^2 > n$. Тогда $\lfloor\sqrt{n}\rfloor = i - 1$. Машина M_f сотрёт с ленты n и i^2 и оставит на ленте только ответ.

Теперь оценим время работы м.Т. M. M_0 работает за время $O(n)$. На каждой итерации $i \le \lfloor\sqrt{n}\rfloor + 1$, $i^2 \le (\lfloor\sqrt{n}\rfloor + 1)^2 = O(n)$. Поэтому время проверки условия на i-й итерации составляет $O(n^2)$. Время работы тела цикла на одной итерации — $O(n\sqrt{n})$, так как $M_{\text{коп}}$ оба раза срабатывает за время $O(n\sqrt{n})$, а M_{+1} срабатывает за время $O(n)$. Поэтому полное время одной итерации составляет $O(n^2)$. Число итераций не превосходит $\lfloor\sqrt{n}\rfloor + 2$. Значит, время работы цикла равно $O(n^2\sqrt{n})$. Время работы M_f — $O(n)$. Поэтому время работы м.Т. M равно $O(n^2\sqrt{n})$.

22.20. а) Будем использовать следующие машины Тьюринга (после каждой машины указано время работы).
$M_{\text{коп}}\colon w \Rightarrow w\#w$, где w не содержит $\#$, $t = O(|w|^2)$
$M'_{\text{коп}}\colon |^a * |^b \# |^c * |^d \Rightarrow |^a * |^b \# |^c * |^d \# |^d$, $t = O(a+b+c+d^2)$
$M_\times\colon |^a * |^b \Rightarrow |^{ab}$, $t = O(a^2b^2)$
$M_{\text{зам}}\colon |^a \# |^b \# |^c \Rightarrow |^a * |^b * |^c$, $t = O(a+b+c)$
$E\colon w \Rightarrow w$, $t = O(1)$
$M_+\colon |^a * |^b \Rightarrow |^{a+b}$, $t = O(a+b)$
$M_{\text{div}}\colon |^a \Rightarrow |^{\lfloor a/2\rfloor}$, $t = O(a^2)$
$$\Phi\colon |^a * |^b \Rightarrow \begin{cases} 0, & \text{если } a < b \\ 1 & \text{иначе} \end{cases}$$

Следующая машина Тьюринга вычисляет функцию $f_1(x, y)$.

if Φ **then**

$M_{\text{коп}}$; $M'_{\text{коп}}$;

$\mathbf{par}_{\#}(M_{\times}, M_{\times}, E)$;

$M_{\text{зам}}$;

$\mathbf{par}_{*}(E, M_{\times})$; $M_{\times}$

else

M_{+}; M_{div}

endif

Время работы составляет $O(n^{10})$.

23. Тезис Тьюринга-Чёрча и неразрешимые проблемы

23.5. Все, кроме (а).
а) *Утверждение.* Пусть Π — программа, не содержащая операторов вида $z := z + 1$, и пусть σ — произвольное состояние такое, что $\Pi(\sigma)$ определено. Тогда для любой переменной x существует $y \in Var \cup \{0\}$ такой, что $\Pi(\sigma)(x) = \sigma(y)$.
Утверждение доказывается индукцией по построению программы. Из утверждения следует, что программа $x := x + 1$ не эквивалентна никакой программе без операторов прибавления единицы.
б) Нужно завести новую переменную z и заменить все операторы вида $x := 0$ на $x := z$.
в) Оператор $x := y$ эквивалентен следующей программе.
$x := 0$; **пока** $x < y$ **делай** $x := x + 1$ **всё**
г) Ветвление эквивалентно следующей программе (в ней u, v и x' — новые переменные).
$u := 0; v := 0; x' := x;$
пока $x < y$ **делай** $P_1; x := y; u := u + 1$ **всё**;
$x := x'$;
если $u = v$ **то** P_2 **иначе** $x := x$ **всё**
д) Аналогично предыдущему пункту.
е) Цикл эквивалентен следующей программе (u и v – новые переменные).
$v := 0$;
если $x < y$ **то** $u := 0$ **иначе** $u := 0; u := u + 1$ **конец**;
пока $u = v$ **делай**

P;

если $x < y$ **то** $u := 0$ **иначе** $u := 0; u := u + 1$ **конец**
всё

ж) Аналогично предыдущему пункту.

23.7. а) По произвольной программе П строим программу $\Pi_1 = x := 0;$ П; $y := 0$. П останавливается на нуле тогда и только тогда, когда Π_1 вычисляет константу.

б) По произвольной программе П строим программу $\Pi_1 = x := 0;$ П; $y := 0; \underbrace{y := y+1; \ldots; y := y+1}_{b \text{ раз}}$. П останавливается на нуле тогда и только тогда, когда $\Phi_{\Pi_1, y}(a) = b$.

23.8. а) $c_{A\cap B}(x) = c_A(x)c_B(x)$

б) $c_{A\cup B}(x) = sg(c_A(x) + c_B(x))$

в) $c_{\bar{A}}(x) = 1 \dot{-} c_A(x)$

23.9. $c_{A+B}(z) = sg(\sum\limits_{x=0}^{z}\sum\limits_{y=0}^{z}(c_A(x)c_B(y)\overline{sg}(|(x+y)-z|)))$

23.11. Пусть о.р.ф. f сводит A к C, а о.р.ф. g сводит B к C. Тогда о.р.ф.

$$\varphi(x) = \begin{cases} f(\frac{x}{2}), & \text{если } x \text{ – чётное число} \\ g(\frac{x-1}{2}) & \text{иначе} \end{cases}$$

сводит $A \oplus B$ к C.

23.12. а) Пусть A рекурсивно перечислимо, тогда $A = \rho_f$ для некоторой о.р.ф. Пусть P — множество простых чисел. Если $A \cap P = \varnothing$, то $A \cap P$ рекурсивно перечислимо. Пусть $A \cap P \neq \varnothing$ и $a \in A \cap P$. Определим функцию $g(x)$ следующим образом:

$$g(x) = \begin{cases} f(x), & \text{если } f(x) \text{ – простое число} \\ a & \text{иначе.} \end{cases}$$

Тогда $g(x) = f(x)p(f(x)) + a(1 \dot{-} p(f(x)))$, где $p(x) = 1$, если x — простое число, $p(x) = 0$ в противном случае (см. задачу 21.13). Получаем $\rho_g = A \cap B$.

б) Пусть A и B рекурсивно перечислимы, тогда Пусть $\rho_f = A$, $\rho_g = B$ для некоторых о.р.ф. f и g. Определим функцию $h(x)$ следующим образом:

$$h(x) = \begin{cases} f(\frac{x}{2}), & \text{если } x \text{ – чётное число} \\ g(\frac{x-1}{2}) & \text{иначе.} \end{cases}$$

Тогда $h(x)$ — о.р.ф. и $\rho_h = A \cup B$.

Литература

[1] Виленкин Н.Я. Популярная комбинаторика. — М.: Наука, 1975.

[2] Гаврилов Г.П., Сапоженко А.А. Сборник задач по дискретной математике. Учеб. пособие. — 3-е изд., перераб. — М.: ФИЗМАТЛИТ, 2005.

[3] Гиндикин С.Г. Алгебра логики в задачах.— М.: Наука, 1972.

[4] **Дехтярь М.И. Лекции по дискретной математике: учебное пособие. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007.**

[5] Ерусалимский Я.М. Дискретная математика: теория, задачи, приложения. — М.: Вузовская книга, 2000.

[6] Кристофидес Н. Теория графов. Алгоритмический подход. — М. : Мир, 1978.

[7] Лавров И. А., Максимова Л. Л. Задачи по теории множеств, математической логике и теории алгоритмов. — М.: Наука, 1984.

[8] Мельников О.И. Занимательные задачи по теории графов. Учебно-методическое пособие. — Минск:НТООО "ТетраСистемс 2001.

[9] Новиков Ф.А. Дискретная математика для программистов.— СПб: Питер, 2000.

[10] Соминский И.С., Головина Л.И., Яглом И.М. О математической индукции. — М.: Наука, 1967.

[11] Тишин В. В. Дискретная математика в примерах и задачах. — СПб.: БХВ-Петербург, 2008.

[12] Трахтенброт Б.А. Сложность алгоритмов и вычислений. — Новосибирск: Изд. НГУ, 1967.

[13] Эвнин А.Ю. Задачник по дискретной математике. 2-е изд. — Челябинск: Издательство ЮУрГУ, 2002.

[14] Яблонский С.В. Введение в дискретную математику. — М.: Наука, 1979.

[15] Johnsonbaugh R. Discrete mathematics. 5th ed. — Upper Saddle River: Prentice-Hall, Inc., 2001.

УДК 510.5 + 519.1 (075.8)
ББК В174я73-4
Д 39

Учебное издание

Дехтярь Михаил Иосифович, Карлов Борис Николаевич

ЗАДАЧНИК ПО ДИСКРЕТНОЙ МАТЕМАТИКЕ

Учебное пособие

Подписано в печать 23.10.2013. Уч.-изд.л. 10,75.
Электронное изд. Заказ 403.
Тверской государственный университет
Факультет прикладной математики и кибернетики
Адрес: 170000, г.Тверь, пер.Садовый, 35.