МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САХАЛИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

**для студентов по выполнению
практических работ
по дисциплине**

ЕН.02 ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОЙ ЛОГИКИ

**укрупненная группа:
09.00.00 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА**

специальность: 09.02.03 Программирование в компьютерных системах

базовый уровень подготовки

**Квалификация: техник-программист
Форма обучения: очная**

Южно-Сахалинск
2020

УТВЕРЖДАЮ
Зам. директора по НМР
__________ Е.Н. Ермолаева
«22» мая 2020г.

Разработчик: Ким А.Х., преподаватель информационных дисциплин

Одобрено на заседании ПЦК
информационных дисциплин
Протокол № 9 от «15» мая 2020г.
Председатель ПЦК
__________ / Куприяшина О.Б.

СОДЕРЖАНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Практические задания учебной дисциплины «Элементы математической логики» предназначены для реализации Государственных требований к минимуму содержания и уровню подготовки выпускников в структуре основной профессиональной образовательной программы: является частью основной профессиональной образовательной программы по специальности 09.02.03. Программирование в компьютерных системах, реализующие основные профессиональные образовательные программы среднего профессионального образования.

Практическое занятие это форма организации учебного процесса, предполагающая выполнение студентами по заданию и под руководством преподавателя одной или нескольких Практических работ. Они составляют важную часть профессиональной практической подготовки специалистов.

Цели выполнения практических работ:

- закрепление знаний по теоретическим основам математической логики;
- получение практических навыков решения задач логического характера.

После выполнения практической работы и собеседования по ней с преподавателем студенту выставляется оценка.

При отборе содержания, предлагаемых в пособии практических занятий, преподаватель руководствовался квалификационными требованиями к выпускнику специальности 09.02.03. Программирование в компьютерных системах. Анализ Государственных требований и содержания учебной дисциплины позволил выявить умения, овладение которыми возможно в ходе изучения учебного материала по дисциплине «Элементы математической логики»:

уметь:

- формулировать задачи логического характера и применять средства математической логики для их решения

знать:

- основные принципы математической логики, теории множеств и теории алгоритмов;
- формулы алгебры высказываний;
- методы минимизации алгебраических преобразований;
- основы языка и алгебры предикатов.

Содержание Практических занятий охватывает круг профессиональных умений, на формирование которых ориентирована данная дисциплина.

Практические занятия носят как репродуктивный, так и исследовательский характер. Это позволяет обеспечить высокий уровень познавательной деятельности студентов.

Формы организации занятий могут быть различны.

Практические задания формируют следующие практические умения и навыки и компетенции:

ПК 1.1. Выполнять разработку спецификаций отдельных компонент..

ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.

ПК 2.4. Реализовывать методы и технологии защиты информации в базах данных.

ПК 3.4. Осуществлять разработку тестовых наборов и тестовых сценариев

В процессе освоения дисциплины студент должен овладевать общими компетенциями:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

ПЕРЕЧЕНЬ ТЕМ И ЗАДАНИЙ ДЛЯ ПРАКТИЧЕСКИХ РАБОТ

№ работы	Название темы	Задание для практической работы	Кол-во часов
1.	**Тема 1.2. Таблица истинности. Варианты импликации.**	**Задание_1.** Составить таблицу истинности для заданного логического выражения.	2
2.	**Тема 1.3. Законы логики. Упрощение формул логики с помощью равносильных преобразований.**	**Задание_1.** Упростить логическое выражение. **Задание_2.** Упростить логическое выражение. **Задание_3.** Упростить логическое выражение. **Задание _4.** Упростить логическое выражение.	2
3.	**Тема 1.5. Совершенная дизъюнктивная и совершенная конъюнктивная нормальные формы.**	**Задание_1.** Составить СДНФ функции, заданной таблицей истинности и упростить полученную функцию. **Задание_2.** Составить СКНФ функции, заданной таблицей истинности и упростить полученную функцию.	2
4.	**Тема 1.6. Многочлены Жегалкина.**	**Задание_1.** Получите многочлен Жегалкина из ДНФ **Задание_2.** Получите многочлен Жегалкина с помощью совершенной дизъюнктивной нормальной формы (СДНФ) **Задание_3.** Получите многочлен Жегалкина методом неопределенных коэффициентов.	2
5.	**Тема 2.1. Основы теории множеств.**	**Задание_1.** Решение задач и построение диаграмм Эйлера-Венна .	2
6.	**Тема 2.2. Отношения. Отображения. Функции.**	**Задание_1.** Найти элементы бинарного отношения. Задать отношение *R*. Определить обратное отношение. **Задание_2.** Указать элементы отношения R, записать матрицу отношения R, определить, является ли R отношением эквивалентности. **Задание_3.** Определить свойства бинарного отношения Q, заданного на данном множестве с обоснованием. **Задание_4.** Определить тип заданного отношения W. **Задание_5.** Определить, является ли заданное отношение функцией, если да, то является ли она	2

		тотальной, сюръекцией, инъекцией, биекцией.	
7.	**Тема 3.1. Логика предикатов.**	**Задание_1.** Записать высказывание и определит его истинность, считая, что все переменные принадлежат множеству действительных чисел. **Задание_2.** Записать предложенное высказывание в символичной форме, введя предикаты	2
8.	**Тема 3.2 Формулы логики предикатов.**	**Задание_1.** Найти множества истинности данных предикатов, если их область определения множество всех действительных чисел. **Задание_2.** Определите следующие предикаты и найти их множества истинности.	2
9.	**Тема 3.3 Приведенные и нормальные формы в логике предикатов.**	**Задание_1.** Привести формулы к предваренной нормальной форме.	2
10.	**Тема 3.4 Исчисление предикатов.**	**Задание_1:** Постройте отрицание к высказываниям, содержащим кванторы. **Задание_2:** Проверьте правильность умозаключений..	2
11.	**Тема 4.2. Машина Поста.**	**Задание_1:** Составьте программу для машины Поста.	2
12.	**Тема4.3. Машина Тьюринга.**	**Задание_1:** Составьте программу для машины Тьюринга.	2
13.	**Тема4.4. Нормальные алгоритмы Маркова**	**Задание_1:** Составьте нормальные алгоритмы Маркова для задач.	2
		Итого:	**26**

ПРАКТИЧЕСКАЯ РАБОТА 1

ТЕМА: Построение таблиц истинности логических выражений.
ЦЕЛЬ: Научиться составлять таблицы истинности логических выражений.
Задание 1.
Составить таблицу истинности для заданного логического выражения.

Вариант 1

1. $F = \overline{\overline{A} + B} + A \cdot \overline{B}$
2. $F = A \vee (\overline{B} + C) + \overline{\overline{A}B}$
3. $F = C + A\overline{B} + (\overline{A\overline{B} + A})$
4. $\overline{(X \vee Y)} \vee (Z \to X) \,\&\, (Z \leftrightarrow Y)$
5. $(A \to B) \vee \overline{A}(C \leftrightarrow D)$

Вариант 2

1. $F = (\overline{A} \vee C)\overline{A} + (B \vee \overline{C})$
2. $F = (A \vee B)(\overline{A} + B)$
3. $F = (B \vee \overline{C}) + A \vee \overline{\overline{A}B}$
4. $\overline{((X \vee Z) \,\&\, (Z \leftrightarrow X))} \,\&\, (Z \to Y)$
5. $A \vee \overline{B}(C \to \overline{D})$

Вариант 3

1. $F = \overline{A\overline{B}} + C + \overline{A}$
2. $F = \overline{A \vee \overline{B}} + A \cdot B$
3. $F = \overline{A \vee \overline{B}} + C\overline{A}$
4. $(X \leftrightarrow Z) \,\&\, (\overline{X} \vee X) \,\&\, (Z \vee Y)$
5. $(A \leftrightarrow B)\overline{(C \vee D)}$

Вариант 4

1. $F = C\overline{A} \vee (A + \overline{B})$
2. $F = (\overline{AB} + \overline{C}) \vee B\overline{C}$
3. $F = A + B(\overline{A} \vee \overline{C})$
4. $(X \& Y) \,\&\, \overline{(\overline{X} \vee X)} \,\&\, (Z \leftrightarrow Y)$
5. $\overline{(A \to B)} \vee C\overline{D}$

Вариант 5

1. $F = A\overline{B} \vee \overline{A}C + \overline{C}$
2. $F = \overline{\overline{A}}\,\overline{C} + \overline{C \vee B}$
3. $F = A(\overline{B} + C) + \overline{A} \vee B$
4. $\overline{((X \vee Y) \,\&\, (Z \leftrightarrow X))} \,\&\, (Z \vee Y)$
5. $(A \vee B)\overline{(C\overline{D})}$

Теоретическая часть

Логические операции и таблицы истинности

1) Логическое умножение или конъюнкция:

Конъюнкция - это сложное логическое выражение, которое считается истинным в том и только том случае, когда оба простых выражения являются истинными, во всех остальных случаях данное сложенное выражение ложно.

Обозначение: F = A & B.

Таблица истинности для конъюнкции

A	**B**	**F**
1	1	1
1	0	0
0	1	0
0	0	0

2) Логическое сложение или дизъюнкция:

Дизъюнкция - это сложное логическое выражение, которое истинно, если хотя бы одно из простых логических выражений истинно и ложно тогда и только тогда, когда оба простых логических выраженья ложны.

Обозначение: F = A v B.

Таблица истинности для дизъюнкции

A	**B**	**F**
1	1	1
1	0	1
0	1	1
0	0	0

3) Логическое отрицание или инверсия:

Инверсия - это сложное логическое выражение, если исходное логическое выражение истинно, то результат отрицания будет ложным, и наоборот, если исходное логическое выражение ложно, то результат отрицания будет истинным. Другими простыми слова, данная операция означает, что к исходному логическому выражению добавляется частица НЕ или слова НЕВЕРНО, ЧТО.

Обозначение: F = ¬A.

Таблица истинности для инверсии

A	**¬A**
1	0
0	1

4) Логическое следование или импликация:

Импликация - это сложное логическое выражение, которое истинно во всех случаях, кроме как из истины следует ложь. То есть данная логическая операция связывает два простых логических выражения, из которых первое является условием (A), а второе (B) является следствием.

«A → B» истинно, если из A может следовать B.

Обозначение: F = A → B.

Таблица истинности для импликации

A	B	F
1	1	1
1	0	0
0	1	1
0	0	1

5) Логическая равнозначность или эквивалентность:

Эквивалентность - это сложное логическое выражение, которое является истинным тогда и только тогда, когда оба простых логических выражения имеют одинаковую истинность.

«A ↔ B» истинно тогда и только тогда, когда А и B равны.

Обозначение: F = A ↔ B.

Таблица истинности для эквивалентности

A	B	F
1	1	1
1	0	0
0	1	0
0	0	1

6) **Операция XOR (исключающие или)**

«A ⊕ B» истинно тогда, когда истинно А или B, но не оба одновременно. Эту операцию также называют "сложение по модулю два".

Обозначение: F = A ⊕ B.

A	B	F
1	1	0
1	0	1
0	1	1
0	0	0

Порядок выполнения логических операций в сложном логическом выражении

1. Инверсия;
2. Конъюнкция;
3. Дизъюнкция;
4. Импликация;
5. Эквивалентность.

Для изменения указанного порядка выполнения логических операций используются скобки.

Пример 1. Определите, какой является формула - выполнимой, опровержимой, тождественно-истинной, тождественно-ложной?

$$\big((x \vee \overline{y}) \rightarrow\big)(\overline{x} \vee y)$$

Решение:

1. Определить количество строк:

на входе два простых высказывания: x, y, поэтому n=2 и количество строк = 2, 2^2 +1 =5.

2. Определить количество столбцов:

– простые выражения (переменные): x, y;

– промежуточные результаты (логические операции):

1) - $\overline{x}$ инверсия;
2) – $\overline{y}$ инверсия;
3) $(x \vee \bar{y})$ – дизъюнкции, т.е. $(x \vee \boxed{2})$;
4) $((x \vee \bar{y}) \rightarrow y)$ - импликация, т.е. $(\boxed{3} \rightarrow y)$;
5) $(\bar{x} \vee y)$ - дизъюнкция, т.е. $(\boxed{1} \vee y)$;
6) $((x \vee \bar{y}) \rightarrow y) \wedge (\bar{x} \vee y)$ - конъюнкция, т.е. $(\boxed{4} \wedge \boxed{5})$, это окончательное значение логического выражения.

3. Заполнить столбцы с учетом таблиц истинности логических операций.

x	y	1	2	3	4	5	6
0	0	1	1	1	0	1	0
0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	0
1	1	0	0	1	1	1	1

Ответ: Формула не является ни тождественно-истинной, ни тождественно-ложной, она выполнима и опровержима.

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 2

ТЕМА: Упрощение логических выражений с помощью равносильных преобразований.
ЦЕЛЬ: Научиться упрощать логические выражения.

Вариант 1

Задание 1.
Упростить логическое выражение.

$$xy(\bar{z} \vee x) \vee z(\bar{x} \cdot \bar{y} \vee \bar{z} \cdot \bar{y}) \vee \bar{x}$$

Задание 2.
Упростить логическое выражение.

$$a(\overline{b} \cdot \overline{a} \vee bac) \vee c\overline{a}(b \vee a)$$

Задание 3.
Упростить булеву функцию.

$$F(p,q,r) = (p \downarrow (q \oplus r)) \wedge (\bar{r} \mid \bar{p}) \Rightarrow \bar{q}$$

Задание 4.
Упростить булеву функцию.

$$F(x,y,z) = \bar{z} \Rightarrow (\bar{y} \Leftrightarrow x) \wedge (x \Rightarrow (\bar{z} \vee y))$$

Вариант 2

Задание 1.
Упростить логическое выражение.

$$xy \vee \bar{z}(x \vee z)(y \vee \bar{y}x) \vee \bar{x} \cdot \bar{z}$$

Задание 2.
Упростить логическое выражение.

$$(a \vee b\overline{c})(\overline{b} \cdot \overline{a} \vee \overline{c}) \vee (b \vee c)(\overline{a} \vee b)$$

Задание 3.
Упростить булеву функцию.

$$F(p,q,r) = (p \mid \bar{r}) \Leftrightarrow (\overline{p \vee q}) \downarrow (r \Rightarrow q)$$

Задание 4.
Упростить булеву функцию.

$$F(x,y,z) = y \Leftrightarrow (\bar{x} \wedge (\overline{y \Leftrightarrow z}) \vee z) \wedge (z \Rightarrow y)$$

Вариант 3

Задание 1.
Упростить логическое выражение.

$$xy\bar{z} \vee \bar{z}x \vee (x\bar{y} \vee z)(\bar{x} \vee zy) \vee x$$

Задание 2.
Упростить логическое выражение.

$$a \vee b\overline{a}(cb \vee \overline{b}) \vee bac \vee \overline{c}(a \vee \overline{b}c)$$

Задание 3.
Упростить булеву функцию.

$$F(p,q,r) = (p \vee (q \oplus r)) \wedge (\bar{r} \downarrow \bar{p}) \Rightarrow \bar{q}$$

Задание 4.
Упростить булеву функцию.

$$F(x,y,z) = \bar{y} \Leftrightarrow ((x \Rightarrow y) \vee (y \Leftrightarrow z) \vee (x \wedge \bar{z}))$$

Вариант 4

Задание 1.
Упростить логическое выражение.

$$\bar{x}(y \vee \bar{z})(\bar{y} \vee xz) \vee x$$

Задание 2.
Упростить логическое выражение.

$$(a \vee c\bar{b})(\bar{b} \vee \bar{c}) \vee bac(\bar{a} \vee acb)$$

Задание 3.
Упростить булеву функцию.

$$F(a,b,c) = ((b \wedge c) \mid a) \Rightarrow ((c \oplus \bar{b}) \Leftrightarrow \bar{a})$$

Задание 4.
Упростить булеву функцию.

$$F(x,y,z) = (y \Leftrightarrow x) \wedge (x \Rightarrow (\bar{z} \vee \bar{y})) \Rightarrow \bar{x}$$

Теоретическая часть

Равносильные преобразования логических формул имеют то же назначение, что и преобразования формул в обычной алгебре. Они служат для упрощения формул или приведения их к определённому виду путем использования основных законов алгебры логики.

Под упрощением формулы, не содержащей операций импликации и эквиваленции, понимают равносильное преобразование, приводящее к формуле, которая либо содержит по сравнению с исходной меньшее число операций конъюнкции и дизъюнкции и не содержит отрицаний неэлементарных формул, либо содержит меньшее число вхождений переменных.

Некоторые преобразования логических формул похожи на преобразования формул в обычной алгебре (вынесение общего множителя за скобки, использование переместительного и сочетательного законов и т.п.), тогда как другие преобразования основаны на свойствах, которыми не обладают операции обычной алгебры (использование распределительного закона для конъюнкции, законов поглощения, склеивания, де Моргана и др.).

Законы алгебры логики

название	для И	для ИЛИ
двойного отрицания	$\overline{\overline{A}} = A$	
исключения третьего	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
операции с константами	$A \cdot 0 = 0,\ A \cdot 1 = A$	$A + 0 = A,\ A + 1 = 1$
повторения	$A \cdot A = A$	$A + A = A$
поглощения	$A \cdot (A + B) = A$	$A + A \cdot B = A$
переместительный	$A \cdot B = B \cdot A$	$A + B = B + A$
сочетательный	$A \cdot (B \cdot C) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
распределительный	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
законы де Моргана	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

Упрощение логических выражений

Шаг 1. Заменить операции $\oplus \rightarrow \leftrightarrow$ на их выражения через **И**, **ИЛИ** и **НЕ**:

$$A \oplus B = A \cdot \overline{B} + \overline{A} \cdot B$$

$$A \rightarrow B = \overline{A} + B$$

$$A \leftrightarrow B = A \cdot B + \overline{A} \cdot \overline{B}$$

Шаг 2. Раскрыть инверсию сложных выражений по формулам де Моргана:

$$\overline{A \cdot B} = \overline{A} + \overline{B}, \quad \overline{A + B} = \overline{A} \cdot \overline{B}$$

Шаг 3. Используя законы логики, упрощать выражение, стараясь применять закон исключения третьего.

Пример: Упростить логическое выражение.

$(B \rightarrow A) \cdot \overline{(A + B)} \cdot (A \rightarrow C)$

Раскрыли $\rightarrow$

$= (\overline{B} + A) \cdot \overline{(A + B)} \cdot (\overline{A} + C)$

Формула де Моргана

$= (\overline{B} + A) \cdot \overline{A} \cdot \overline{B} \cdot (\overline{A} + C)$

Распределительный

$= (\overline{B} \cdot \overline{A} + A \cdot \overline{A}) \cdot \overline{B} \cdot (\overline{A} + C)$

Исключения третьего

$= \overline{B} \cdot \overline{A} \cdot \overline{B} \cdot (\overline{A} + C)$

Повторения

$= \overline{B} \cdot \overline{A} \cdot (\overline{A} + C)$

Поглощения

$= \overline{B} \cdot \overline{A}$

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 3

ТЕМА: Получение СДНФ и СКНФ по таблицам истинности.
ЦЕЛЬ: Закрепить навык синтеза логических выражений по таблицам истинности.

Задание 1.
Составить СДНФ функции, заданной таблицей истинности и упростить полученную функцию.

А)

x	y	z	F(x,y,z)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Б)

x	y	z	F(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Задание 2.
Составить СКНФ функции, заданной таблицей истинности и упростить полученную функцию.

А

x	y	z	F(x,y,z)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Б)

x	y	z	F(x,y,z)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Теоретическая часть

Элементарной конъюнкцией называется конъюнкция нескольких переменных, взятых с отрицанием или без отрицания, причем среди переменных могут быть одинаковые.

Элементарной дизъюнкцией называется дизъюнкция нескольких переменных, взятых с отрицанием или без отрицания, причем среди переменных могут быть одинаковые.

Всякую дизъюнкцию элементарных конъюнкций назовем **дизъюнктивной нормальной формой (ДНФ)**.

Всякую конъюнкцию элементарных дизъюнкций назовем **конъюнктивной нормальной формой (КНФ)**.

Совершенной дизъюнктивной нормальной формой (СДНФ) называется ДНФ, в которой нет одинаковых элементарных конъюнкций и все конъюнкции состоят из одного и того же набора переменных, в который каждая переменная входит только один раз (возможно, с отрицанием).

Совершенной конъюнктивной нормальной формой (СКНФ) называется КНФ, в которой нет одинаковых элементарных дизъюнкций и все дизъюнкции состоят из одного и того же набора переменных, в который каждая переменная входит только один раз (возможно, с отрицанием).

Алгоритм получения СДНФ по таблице истинности:

1. Отметить те строчки таблицы истинности, в последнем столбце которых стоят 1:

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

2. Выписать для каждой отмеченной строки *конъюнкцию* всех переменных следующим образом: если значение некоторой переменной в данной строке *равно 1*, то в конъюнкцию включать *саму эту переменную*, если *равно 0*, то ее *отрицание*:

A	B	C	X	
0	0	0	1	$\overline{A}\cdot\overline{B}\cdot\overline{C}$
0	0	1	1	$\overline{A}\cdot\overline{B}\cdot C$
0	1	0	1	$\overline{A}\cdot B\cdot\overline{C}$
0	1	1	1	$\overline{A}\cdot B\cdot C$
1	0	0	0	
1	0	1	1	$A\cdot\overline{B}\cdot C$
1	1	0	0	
1	1	1	1	$A\cdot B\cdot C$

3. Все полученные конъюнкции связать в дизъюнкцию:

$$X=\overline{A}\cdot\overline{B}\cdot\overline{C}+\overline{A}\cdot\overline{B}\cdot C+\overline{A}\cdot B\cdot\overline{C}+\overline{A}\cdot B\cdot C+A\cdot\overline{B}\cdot C+A\cdot B\cdot C$$

Алгоритм получения СКНФ по таблице истинности:

1. Отметить те строки таблицы истинности, в последнем столбце которых стоит 0:

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

2. Выписать для каждой отмеченной строки *дизъюнкцию* всех переменных следующим образом: если значение некоторой переменной в данной строке *равно 0*, то в дизъюнкцию включать *саму эту переменную*, если *равно 1*, то ее *отрицание:*

A	B	C	X	
0	0	0	0	$A+B+C$
0	0	1	1	
0	1	0	0	$A+\overline{B}+C$
0	1	1	1	
1	0	0	0	$\overline{A}+B+C$
1	0	1	0	$\overline{A}+B+\overline{C}$
1	1	0	0	$\overline{A}+\overline{B}+C$
1	1	1	1	

3. Все полученные дизъюнкции связать в конъюнкцию:

$$X = (A+B+C)\cdot(A+\overline{B}+C)\cdot(\overline{A}+B+C)\cdot(\overline{A}+B+\overline{C})\cdot(\overline{A}+\overline{B}+C)$$

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 4

ТЕМА: Получение многочлена Жегалкина из ДНФ и таблицы истинности.
ЦЕЛЬ: Закрепить умение получать многочлен Жегалкина различными способами.

Задание 1.
Получите многочлен Жегалкина из ДНФ

А)

$$xy\bar{z} \vee x\bar{z}\bar{y} \vee \bar{z}\bar{y}$$

Б)

$$x \vee y \vee \bar{y}x \vee \bar{x}\bar{z}$$

Задание 2.
Получите многочлен Жегалкина с помощью совершенной дизъюнктивной нормальной формы (СДНФ)

А)

x	y	z	F(x,y,z)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Б)

x	y	z	F(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Задание 3.
Получите многочлен Жегалкина методом неопределенных коэффициентов

А)

x	y	z	F(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Б)

x	y	z	F(x,y,z)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Теоретическая часть

Это - еще один способ выразить произвольную булеву функцию через бинарные операции. Мы уже знаем, что произвольную булеву функцию можно выразить через &, $\vee$ и ~ в виде СДНФ. Мы также знаем, что $\vee$ и ~ можно выразить через & и $\oplus$:

$$x \vee y = x \oplus y \oplus (x \,\&\, y)$$

$$\neg x = x \oplus 1$$

Рассмотрим алгоритмы построения полинома Жегалкина булевой функции, заданной различными способами, а именно: совершенной ДНФ, произвольной ДНФ, формулой и таблицей истинности.

Алгоритм построения полинома Жегалкина по СДНФ (основан на доказательстве теоремы о существовании полинома Жегалкина).

Начало. Задана совершенная ДНФ функции $f(x_1, \ldots, x_n)$.

Шаг 1. Заменяем каждый символ дизъюнкции на символ дизъюнкции с исключением.

Шаг 2. Заменяем каждую переменную с инверсией x равносильной формулой x $\oplus$ 1.

Шаг 3. Раскрываем скобки.

Шаг 4. Вычеркиваем из формулы пары одинаковых слагаемых.

Конец. Получен полином Жегалкина функции $f(x_1, \ldots, x_n)$.

Пример. Найдем полином Жегалкина мажоритарной булевой функции по ее совершенной ДНФ.

$$
\begin{gathered}
\text{СовДНФ} = \overline{x}yz \vee x\overline{y}z \vee xy\overline{z} \vee xyz = \\
= \overline{x}yz \oplus x\overline{y}z \oplus xy\overline{z} \oplus xyz = \\
= (1 \oplus x)yz \oplus x(1 \oplus y)z \oplus xy(1 \oplus z) \oplus xyz = \\
= yz \oplus \cancel{xyz} \oplus xz \oplus \cancel{xyz} \oplus xy \oplus \cancel{xyz} \oplus \cancel{xyz} = \\
= yz \oplus xz \oplus xy = P. \bullet
\end{gathered}
$$

Алгоритм построения полинома Жегалкина по ДНФ (основан на равносильности $K_1 \vee K_2 = K_1 \oplus K_2 \oplus K_1K_2$).

Начало. Задана произвольная ДНФ функции $f(x_1, \ldots, x_n)$.

Шаг 1. Разбиваем ДНФ на пары конъюнкций, предпочтительно ортогональных (если число конъюнкций нечетно, одна из них остается без пары).

Шаг 2. Заменяем дизъюнкцию каждой пары конъюнкций $K_1 \vee K_2$ формулой $K_1 \oplus K_2 \oplus K_1K_2$ или формулой $K_1 \oplus K_2$, если K_1 и K_2 ортогональны.

Шаг 3. В полученной формуле находим очередную дизъюнкцию $A_1 \vee A_2$и заменяем ее формулой $A_1 \oplus A_2 \oplus A_1A_2$. Повторяем шаг 3 до тех пор, пока это возможно.

Шаг 4. Заменяем каждую переменную с инверсией x равносильной формулой x $\oplus$ 1.

Шаг 5. Раскрываем скобки.

Шаг 6. Вычеркиваем из формулы пары одинаковых слагаемых.

Конец. Получен полином Жегалкина функции $f(x_1, \ldots, x_n)$.

Пример. Найдем полином Жегалкина мажоритарной функции по ДНФ.

$$
\begin{gathered}
\text{ДНФ} = xy\overline{z} \vee xz \vee yz = (xy\overline{z} \vee xz) \vee yz = \\
= (xy\overline{z} \oplus xz) \vee yz = (xy\overline{z} \oplus xz) \oplus yz \oplus (xy\overline{z} \oplus xz)yz = \\
= xy\overline{z} \oplus xz \oplus yz \oplus xyz = xy(1 \oplus z) \oplus xz \oplus yz \oplus xyz = \\
= xy \oplus \cancel{xyz} \oplus xz \oplus yz \oplus \cancel{xyz} = xy \oplus xz \oplus yz = P. \bullet
\end{gathered}
$$

Отметим, что полиномы мажоритарной функции, полученные в двух последних примерах, совпадают с точностью до порядка конъюнкций, и это естественно (по теореме о единственности полинома Жегалкина).

Алгоритмы построения полинома Жегалкина по формуле.

Способ 1 основан на предварительном преобразовании формулы в ДНФ (любым известным нам способом). Затем ДНФ преобразуется в полином Жегалкина по только что изученному алгоритму.

Примеры. Получим полиномы Жегалкина двух элементарных булевых функций: импликации и эквивалентности, представив их предварительно кратчайшими ДНФ.

$$
\begin{gathered}
a \to b = \overline{a} \vee b = \overline{a} \oplus b \oplus \overline{a}b = (1 \oplus a) \oplus b \oplus (1 \oplus a)b = \\
= 1 \oplus a \oplus \cancel{b} \oplus \cancel{b} \oplus ab = 1 \oplus a \oplus ab; \\
a \sim b = \overline{a}\,\overline{b} \vee ab = \overline{a}\,\overline{b} \oplus ab = (1 \oplus a)(1 \oplus b) \oplus ab = \\
= 1 \oplus a \oplus b \oplus \cancel{ab} \oplus \cancel{ab} = 1 \oplus a \oplus b. \bullet
\end{gathered}
$$

Аналогично можно получить полиномы Жегалкина всех элементарных булевых функций (оставим читателю их вывод).

$$
\begin{array}{ll}
a \vee b = a \oplus b \oplus ab & a \rightarrow b = 1 \oplus a \oplus ab \\
a \sim b = 1 \oplus a \oplus b & a \leftarrow b = 1 \oplus b \oplus ab \\
a \downarrow b = 1 \oplus a \oplus b \oplus ab & a \hookrightarrow b = a \oplus ab \\
a / b = 1 \oplus ab & a \hookleftarrow b = b \oplus ab
\end{array}
$$

Константы 0 и 1, тождественная функция, а также конъюнкция ab и дизъюнкция с исключением $a \oplus b$ уже являются полиномами Жегалкина. Полином Жегалкина инверсии $a=1 \oplus a$.

Заметим, что некоторые из приведенных полиномов могут быть получены гораздо проще, в частности,

$$a \sim b = \overline{a \oplus b} = 1 \oplus a \oplus b.$$

Способ 2. Если булева функций задана произвольной формулой, то ее полином Жегалкина можно получить подстановкой в формулу вместо элементарных булевых функций их полиномов.

Пример. Найдем полином Жегалкина мажоритарной функции, заданной формулой:

$F=x \sim y \longleftarrow z / (x \longrightarrow y) = ((x \sim y) \longleftarrow z) / (x \longrightarrow y) =$

[подставим в формулу полином Жегалкина штриха Шеффера $1 \oplus ab$ при $a=(x \sim y) \longleftarrow z$, $b=x \longrightarrow y$]

$= 1 \oplus ((x \sim y) \longleftarrow z)\ (x \longrightarrow y) =$

[подставим полиномы Жегалкина обратной импликации $1 \oplus b \oplus ab$ при $a=x \sim y$, $b=z$ и импликации $1 \oplus a \oplus ab$ при $a=x$, $b=y$]

$= 1 \oplus (1 \oplus z \oplus (x \sim y)z)\ (1 \oplus x \oplus xy) =$

[подставим полином Жегалкина эквивалентности $1 \oplus x \oplus y$, раскроем скобки, и вычеркнем появившиеся при этом пары одинаковых слагаемых]

$$
\begin{gathered}
= 1 \oplus (1 \oplus z \oplus (1 \oplus x \oplus y) z)(1 \oplus x \oplus x\overline{y}) = \\
= 1 \oplus (1 \oplus \cancel{z} \oplus \cancel{z} \oplus xz \oplus yz)(1 \oplus x \oplus x\overline{y}) = \\
= 1 \oplus (1 \oplus xz \oplus yz)(1 \oplus x \oplus x\overline{y}) = \\
= \cancel{1} \oplus \cancel{1} \oplus \cancel{xz} \oplus yz \oplus x \oplus \cancel{xz} \oplus xyz \oplus x\overline{y} \oplus x\overline{y}z =
\end{gathered}
$$

[заменим инверсию ее полиномом Жегалкина, раскроем скобки и вычеркнем пары одинаковых слагаемых]

$$
\begin{gathered}
= yz \oplus x \oplus xyz \oplus x(1 \oplus y) \oplus x(1 \oplus y)z = \\
= yz \oplus \cancel{x} \oplus \cancel{xyz} \oplus \cancel{x} \oplus xy \oplus xz \oplus \cancel{xyz} = \\
= yz \oplus xy \oplus xz = P. \bullet
\end{gathered}
$$

Полином, естественно, совпадает с полученными ранее по совершенной и произвольной ДНФ.

Способ 3. Если булева функций задана произвольной формулой, то ее полином Жегалкина можно получить, используя специальное разложение функции.

Определение. *Разложением Дэвио* называется следующее разложение булевой функции $f(x_1, \ldots, x_n$ по переменной x_i:

$f(x_1, \ldots, x_n) = x_i f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \oplus$

$\oplus (1 \oplus x_i) f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n).$

Разложение Дэвио непосредственно следует из разложения Шеннона, если учесть, что слагаемые в последнем ортогональны, и что $x_i = x_i \oplus 1$.

Пример. Найдем разложение Дэвио по переменной x мажоритарной булевой функции, заданной формулой.

$F = ((x \sim y) \longleftarrow z) /(x \longrightarrow y)=$

$=x[((1 \sim y) \longleftarrow z) /(1 \longrightarrow y)] \oplus (1 \oplus x) [((0 \sim y) \longleftarrow z) /(0 \longrightarrow y)]=$

$=x[(y \longleftarrow z) /y] \oplus (1 \oplus x)[y \hookleftarrow z].$

Для получения полинома Жегалкина необходимо продолжить разложение подформул, не являющихся дизъюнкцией с исключением элементарных конъюнкций, пока не получится формула над $\{\oplus, \wedge, ^{-}\}$. Если в такой формуле заменить инверсии x на $x \oplus 1$, раскрыть скобки и вычеркнуть пары одинаковых слагаемых, то получится полином Жегалкина.

Пример. Продолжив предыдущий пример, получим полином Жегалкина мажоритарной функции. Для этого разложим подформулы $(y \longleftarrow z)\ /y$ и $y \hookleftarrow z$ по переменной y:

$$
\begin{gathered}
x\,[(y \leftarrow z)\,/\,\overline{y}] \oplus (1 \oplus x)\,[\overline{y} \hookleftarrow z] = \\
= x\,[y\,[(1 \leftarrow z)\,/\,0] \oplus (1 \oplus y)\,[(0 \leftarrow z)\,/\,1]\,] \oplus \\
\oplus (1 \oplus x)\,[y\,[0 \hookleftarrow z] \oplus (1 \oplus y)\,[1 \hookleftarrow z]\,] = \\
= x\,[y \oplus (1 \oplus y)\,z] \oplus (1 \oplus x)\,[y\,z] = \\
= x\,y \oplus x\,z \oplus \cancel{x\,y\,z} \oplus y\,z \oplus \cancel{x\,y\,z} = x\,y \oplus x\,z \oplus y\,z = P. \bullet
\end{gathered}
$$

Полином Жегалкина, естественно, совпадает с полученными ранее.

Алгоритм построения полинома Жегалкина по таблице истинности (основан на методе неопределенных коэффициентов).

Продемонстрируем идею метода на примере произвольной булевой функции двух аргументов f(x,y). Представим ее полиномом Жегалкина в форме с коэффициентами

$P_f = c_0 \oplus c_1 y \oplus c_2 x \oplus c_3 x\ y.$

Подставив в данное равенство наборы значений аргументов, получим систему из четырех линейных уравнений с четырьмя неизвестными коэффициентами: c_0, c_1 c_2, c_3.

$f(0,0) = c_0 \oplus c_1 0 \oplus c_2 0 \oplus c_3 0\ 0 = c_0$

$f(0,1) = c_0 \oplus c_1 1 \oplus c_2 0 \oplus c_3 0\ 1 = c_0 \oplus c_1$

$f(1,0) = c_0 \oplus c_1 0 \oplus c_2 1 \oplus c_3 1\ 0 = c_0 \oplus c_2$

$f(1,1) = c_0 \oplus c_1 1 \oplus c_2 1 \oplus c_3 1\ 1 = c_0 \oplus c_1 \oplus c_2 \oplus c_3$

Заметим, что наборы подставлены в равенство в естественном порядке, и система имеет треугольный вид: в первом уравнении обратились в ноль все слагаемые, следующие за c_0, во втором – следующие за c_1 и так далее. Значит, коэффициент c_0 можно получить из первого уравнения и подставить его в остальные. Тогда c_1 можно получить из второго уравнения, и так далее.

В общем случае для функции *n* аргументов получается система треугольного вида из 2^n линейных уравнений с 2^n неизвестными – коэффициентами полинома Жегалкина.

Пример. Найдем полином Жегалкина мажоритарной булевой функции, заданной таблицей истинности, последовательно вычисляя коэффициенты полинома и подставляя их в остальные уравнения.

$x\ y\ z$	$f = c_0 \oplus c_1 z \oplus c_2 y \oplus c_3 yz \oplus c_4 x \oplus c_5 xz \oplus c_6 xy \oplus c_7 xyz$
$0\ 0\ 0$	$0 = c_0$
$0\ 0\ 1$	$0 = 0 \oplus c_1$
$0\ 1\ 0$	$0 = 0 \oplus 0 \oplus c_2$
$0\ 1\ 1$	$1 = 0 \oplus 0 \oplus 0 \oplus c_3$
$1\ 0\ 0$	$0 = 0 \oplus 0 \oplus 0 \oplus 100 \oplus c_4$
$1\ 0\ 1$	$1 = 0 \oplus 0 \oplus 0 \oplus 101 \oplus 0 \oplus c_5$
$1\ 1\ 0$	$1 = 0 \oplus 0 \oplus 0 \oplus 110 \oplus 0 \oplus 110 \oplus c_6$
$1\ 1\ 1$	$1 = 0 \oplus 0 \oplus 0 \oplus 111 \oplus 0 \oplus 111 \oplus 111 \oplus c_7$

Из первого уравнения следует, что c_0=0. Из второго и третьего уравнений следует, что c_1=0 и c_2=0, значит, c_1z и c_2y тождественно равны нулю. Из четвертого уравнения получаем c_3=1, значит, надо вычислять значения конъюнкции c_3yz в остальных уравнениях. Аналогично получаем c_4=0, c_5=1, c_6=1 и c_7=0. Найден вектор коэффициентов полинома Жегалкина мажоритарной функции π=00010110, и сам полином $P=yz \oplus xz \oplus xy$, который, естественно, совпадает с полученными ранее.

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 5

ТЕМА: Диаграммы Эйлера-Венна. Алгебра множеств.
ЦЕЛЬ: Научится решать задачи на множества с помощью диаграмм Эйлера-Венна.

Задание 1.
Решите задачи и изобразите их на диаграмме Венна.

1. На 20 % компьютеров компании установлена операционная система Microsoft Windows XP , на 85 % компьютеров установлена Microsoft Windows 7, на 10 % установлена операционная система Linux. Одновременно Linux и Microsoft Windows 7 установлены на 6% компьютеров, Microsoft Windows XP и Linux на 4% , все три программы установлены на 2% компьютеров. На скольких процентах компьютеров установлена операционная система Microsoft?
2. Каждый студент в группе сдает экзамен либо по высшей математике, либо по математической логике, либо по обоим предметам. По высшей математике сдают экзамен 15 человек, а по мат. логике - 19, а тот и другой предмет – 7 студентов. Сколько студентов в группе?
3. В торговый центр “Форум” пришло 100 покупателей. Диск Николая Баскова купило 20 человек, диск Стаса Михайлова купило 64 человек, причем 11 человек купило диски этих двух исполнителей. Сколько человек не купило диски этих исполнителей?
4. Несколько футбольных болельщиков соседнего дома выписывают журнал “Наш футбол”, часть жителей этого дома выписывают известный автомобильный журнал “Top Gear”, а часть тот, и тот журнал. Сколько жителей соседнего дома выписывают оба журнала, если на “Наш Футбол” подписано 64 процента, а на “Top Gear” – 84 процента?
5. Первый и второй зачет по Русскому языку сдали 9 школьников, первый и третий зачет – 6 школьников, второй и третий - 7 школьников. Не менее двух зачетов выполнили 10 школьников. Сколько школьников успешно сдали все три зачета?
6. В кондитерском отделе супермаркета посетители обычно покупают либо один торт, либо одну коробку конфет, либо один торт и одну коробку конфет. В один из дней было продано 57 тортов и 36 коробок конфет. Сколько было покупателей, если 12 человек купили и торт, и коробку конфет?
7. В Хоккейной команде “Звезда” 24 игрока. Среди них 13 нападающих. 7 полузащитников, 10 защитники и вратари. Известно, что 4 из игроков могут быть нападающими и защитниками, 5 защитниками и полузащитниками, 7 нападающими и защитниками, а 2 и нападающими и защитником, и полузащитником. Вратари не заменимы. Сколько в команде “Звезда” вратарей?
8. В магазин «Мир музыки» пришло 35 покупателей. Из них 20 человек купили новый диск певицы Максим, 11 – диск Земфиры, 10 человек не купили ни одного диска. Сколько человек купили диски и Максим, и Земфиры?
9. На полке стояло 42 волшебные книги по заклинаниям, все они были прочитаны. Из них 5 прочитали и Гарри Поттер, и Рон. Гермиона прочитала 27 книг, которых не читали ни Гарри Поттер, ни Рон, и 6 книг, которые читал Гарри Поттер. 4 книги прочитали и Рон, и Гермиона. 2 книги прочитали все трое. Всего Гарри Поттер прочитал 11 книг. Сколько книг прочитал только Рон?
10. В магазине побывало 36 человек. Известно, что они купил 10 планшетов, 15 смартфонов, 23 телевизора. 7 из них купило и планшет, и смартфон, 15 человек купили и смартфон, и телевизор, 6 человек – и планшет, и телевизор. И 5 человек совершили все три покупки. Был ли среди них посетитель, который ничего не купил?
11. В офисе работает 119 человек. 25 человек приезжает только на личном авто. Автобусом пользуется 27 человек, троллейбусом 43, метро 36, причем, четверо из

них пользуются и метро и автобусом, 5 человек - троллейбусом и метро, 6 человек - автобусом и троллейбусом. Часть из них пользуются троллейбусом, метро и автобусом. Сколько человек пользуется не одним видом транспорта?

12. Из 100 туристов отправляющихся на зимний курорт, на сноуборде умеют кататься 30 человек, на лыжах – 28 и на коньках – 42 человека. На сноуборде и лыжах умеют кататься 8 человек, на лыжах и на коньках – 5 человек, на сноуборде и коньках 4 человека. На всех трех – трое. Сколько человек вообще не умеет кататься?
13. Каждый из 35 шестиклассников является читателем, по крайней мере, одной из двух библиотек: школьной и районной. Причем, 25 человек из них берут книги в школьной библиотеке, 20 человек берут книги в районной библиотеке. Сколько шестиклассников: 1)не являются читателями районной библиотеки; 2) не являются читателями школьной библиотеки; 3) являются читателями только районной библиотеки; 3) являются читателями только школьной библиотеки? 4) являются читателями обеих библиотек?
14. Из сотрудников фирмы 16 побывали во Франции, 10-в Италии, 6-в Англии; в Англии и Италии -5; в Англии и Франции - 6; во всех трех странах - 5 сотрудников. Сколько человек посетили и Италию, и Францию, если всего в фирме работают 19 человек, и каждый из них побывал хотя бы в одной из названных стран?
15. В трёх группах 70 студентов. Из них 27 занимаются в драмкружке, 32 поют в хоре, 22 увлекаются спортом. В драмкружке 10 студентов из хора, в хоре 6 спортсменов, в драмкружке 8 спортсменов; 3 спортсмена посещают и драмкружок и хор. Сколько студентов не поют в хоре, не увлекаются спортом и не занимаются в драмкружке? Сколько студентов заняты только спортом?

Теоретическая часть

Решить задачу, используя диаграмму Эйлера-Венна.

Четырнадцать спортсменов участвовали в кроссе, 16 – в соревнованиях по плаванию, 10 – в велосипедных гонках. Восемь участников участвовали в кроссе и заплыве, 4 – в кроссе и велосипедных гонках, 9 – в плавании и велосипедных гонках. Во всех трех соревнованиях участвовали три человека. Сколько всего было спортсменов?

Решение:

Универсальное множество U – это множество всех спортсменов, участвовавших в соревнованиях.

Множество К – множество спортсменов, участвовавших в кроссе,

n(К) = 14 – количество элементов множества К.

Множество П – множество спортсменов, участвовавших в соревнованиях по плаванию, n(П) = 16 – количество элементов множества П.

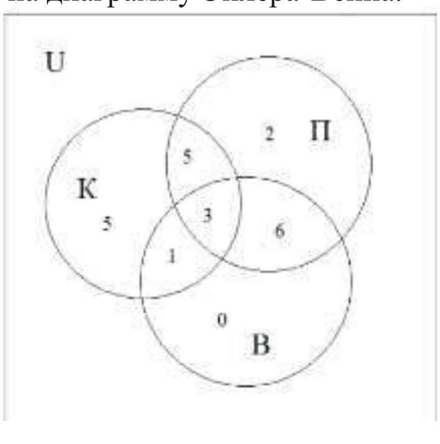
Множество В – множество спортсменов, участвовавших в велосипедных гонках, n(В) = 10 – количество элементов множества В.

Условие задачи:

$n(К) = 14$; $n(П) = 16$; $n(В) = 10$; $n(К \cap П) = 8$; $n(К \cap В) = 4$; $n(П \cap В) = 9$; $n(К \cap П \cap В) = 3$.

Требуется найти n(U).

Перенесем эти данные на диаграмму Эйлера-Венна.



На диаграмме все элементы учтены ровно по одному разу, следовательно, общее количество спортсменов, участвовавших в соревнованиях, равно:

$n(U) = 5+5+2+1+3+6+0 = 22.$

Ответ: Общее число спортсменов равно 22 человека.

Критерии оценки:

Оценка «5» ставится студенту, правильно выполнившему 14-15 задач.

Оценка «4» ставится студенту, правильно выполнившему 10-13 задач.

Оценка «3» ставится студенту, правильно выполнившему 6-9 задач.

Оценка «2» ставится студенту, выполнившему задание менее 6-ти задач.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 6

ТЕМА: Бинарные операции на множестве.
ЦЕЛЬ: научиться записывать бинарные отношения, определять их свойства и типы, определять функции и их типы.

Задание_1.

Найти элементы бинарного отношения $R = \{(a,b)|b$ кратно $a\}$ из множества A во множество B. Отношение R задать с помощью перечисления пар и матрицей. Определить обратное отношение.

1 A = {2; 3; 9; 15; 16}; B = {20; 36; 45; 64}	2 A = {2; 6; 7; 11; 20}; B = {20; 36; 42; 140}	3 A = {2; 3; 6; 7; 12}; B = {6; 16; 42; 60}
4 A = {1; 4; 9; 10; 12}; B = {20; 36; 45; 72}	5 A = {1; 2; 3; 10; 15}; B = {3; 8; 30; 90}	6 A = {2; 3; 9; 11; 17}; B = {6; 18; 44; 51}
7 A = {1; 4; 7; 20; 25}; B = {20; 36; 50; 125}	8 A = {2; 3; 7; 8; 13}; B = {20; 39; 42; 72}	9 A = {2; 5; 6; 7; 10}; B = {6; 16; 40; 140}
10 A = {3; 4; 5; 11; 12}; B = {20; 36; 55; 72}	11 A = {1; 4; 6; 12; 18}; B = {3; 30; 36; 84}	12 A = {5; 6; 9; 15; 18}; B = {6; 18; 45; 75}

Задание_2.

На множестве A задано бинарное отношение R. Указать элементы отношения R, записать матрицу отношения R, определить, является ли R отношением эквивалентности.

1 {2, 3, 4, 5} R={(a,b) a <b }	2 A={12, 13, 14, 15} R={(a, b) a >b}
3 A={1, 2, 3, 4} R={(a, b) a ≤b}	4 A={6, 7, 8, 9} R={(a, b) a ≥ b}
5 A={2, 3, 4, 7} R={(a, b) a кратно b}	6 A={5, 6, 10, 18} R={(a, b) b кратно a}
7 A={2, 4, 6, 8} R={(a, b) a : b -четное }	8 A={1, 2, 4, 6} R={(a, b) a : b - четное}
9 A={1, 2, 3, 4} R={(a, b) a–b<1}	10 A={2, 4, 16, 22} R={(a, b) (a + b) кратно 6}
11 A={2, 4, 8, 10} R={(a, b) (a–b) крат-но 3}	12 A={2, 3, 4, 5} R={(a, b) b - a <1}

Задание_3.

Определить свойства бинарного отношения Q, заданного на данном множестве с обоснованием.

1 Отношение «не равно» на множест-ве действительных чисел	2 Отношение «больше» на множестве действительных чисел
3 Отношение «меньше или равно» на множестве действительных чисел	4 Отношение «больше или равно» на множестве действительных чисел
5 Отношение «остатком от деления нацело» на множестве действительных чисел	6 Отношение «быть делителем» на множестве действительных чисел
7 Отношение «подобия» на множестве треугольников	8 Отношение «равенства» на множе-стве треугольников
9 Отношение «быть старше» на мно-жестве людей	10 Отношение «быть руководителем» на множестве людей
11 Отношение «быть однокурсником» на множестве людей	12 Отношение «быть соседом» на множестве людей

Задание_4.

Определить тип заданного отношения *W*.

1 Отношение «равенства» на множе- стве многоугольников	2 Отношение «подобия» на множестве многоугольников
3 Отношение «ортогональности» на множестве векторов на плоскости	4 Отношение «коллинеарности» на множестве векторов
5 Отношение «быть руководителем» на множестве людей	6 Отношение «быть старше» на мно- жестве людей
7 Отношение «быть соседом» на мно- жестве людей	8 Отношение «быть однокурсником» на множестве людей
9 Отношение «больше или равно» на множестве действительных чисел	10 Отношение «меньше или равно» на множестве действительных чисел
11 Отношение «быть делителем» на множестве действительных чисел	12 Отношение «обучаться на одном факультете» на множестве студентов

Задание_5.

Определить, является ли заданное отношение функцией, если да, то является ли она тотальной, сюръекцией, инъекцией, биекцией

1 {(a, 1), (c, 2), (f, 3), (k, 3)} $f_x = \{a; c; f; k\}, f_y = \{1; 2; 3\}$	*2 {(a, 1), (c, 2), (f, 3), (f, 4)}* $f_x = \{a; c; f; k\}, f_y = \{1; 2; 3; 4\}$
3 {(1, 1), (1, 2), (2, 3), (4, 3)} $f_x = \{1; 2; 3; 4\}, f_y = \{1; 2; 3\}$	*4 {(1, 1), (2, 2), (3, 3), (4, 3), (5,1)}* $f_x = \{1; 2; 3; 4; 5\}, f_y = \{1; 2; 3\}$
5 {(a, ж), (c, ш), (f, ы), (k, э)} $f_x = \{a; c; f; k\}, f_y = \{ж; ш; ы; э; я\}$	*6 {(a, ж), (c, ж), (f, ы), (k, я)}* $f_x = \{a; c; f; k\}, f_y = \{ж; ш; ы; э; я\}$
7 {(1, 10), (2, 10), (3, 20), (4, 30)} $f_x = \{1; 2; 3; 4\}, f_y = \{10; 20; 30\}$	*8 {(1, 1), (2, 2), (3, 3), (4, 3), (5,1)}* $f_x = \{1; 2; 3; 4; 5\}, f_y = \{1; 2; 3\}$
9 {(5, a), (7, m), (12, m), (17, p)} $f_x = \{5; 7; 12; 17\}, f_y = \{a; m; p; q\}$	*10 {(12, a), (5, m), (7, q), (17, p)}* $f_x = \{5; 7; 12; 17\}, f_y = \{a; m; p; q\}$
11 {(11, 12), (23, 6), (35, 6), (49, 22)} $f_x = \{11; 23; 35; 49\}, f_y = \{6; 12; 18; 22\}$	*12 {(11, 18), (11, 6), (35, 12), (49, 22)}* $f_x = \{11; 23; 35; 49\}, f_y = \{6; 12; 18; 22\}$

Теоретическая часть

Основные понятия.

1 Пусть A и B – два множества. Бинарным отношением из множества A во множество B называется любое подмножество R множества $A \times B$: $R \subseteq A \times B$. Если $(a,b) \in R$, то пишут aRb и говорят, что a и b находятся в отношении R. Если $A=B$, то говорят, что R – отношение на множестве A.

2 На конечных множествах бинарное отношение удобно задавать матрицей. Пусть $A=\{a_1,a_2,\ldots,a_m\}$, $B=\{b_1,b_2,\ldots,b_n\}$, $R \subseteq A \times B$. Матрицей $P=(p_{ij})$ бинарного отношения R называется матрица размера $m \times n$, элементы которой определяются так:

$$p_{ij} = \begin{cases} 1, \text{если } (a_i, b_j) \in R \\ 0, \text{если } (a_i, b_j) \notin R \end{cases}$$

3 Областью определения R_A бинарного отношения R называется подмножество множества A, такое, что $R_A = \{a \in A \mid (\exists b \in B) : (a,b) \in R\}$. Областью значений R_B бинарного отношения R называется подмножество множества B, такое, что $R_B = \{b \in B \mid (\exists a \in A) : (a,b) \in R\}$.

4 Отношением R^{-1}, обратным к отношению R, называется подмножество декартового произведения $B \times A$, такое, что $R^{-1} = \{(b,a) \mid (a,b) \in R\}$. Матрица обратного отношения получается транспонированием матрицы отношения R.

5 Пусть R – бинарное отношение на множестве A, т.е. $R \subseteq A \times A$. Отношение R называется:

– рефлексивным, если $(\forall a \in A) : (a,a) \in R$;
– антирефлексивным, если $(\forall a \in A) : (a,a) \notin R$;
– симметричным, если $(\forall a,b \in A, a \neq b) : (a,b) \in R \Rightarrow (b,a) \in R$;
– антисимметричным, если $(\forall a,b \in A) : ((a,b) \in R, (b,a) \in R) \Rightarrow a = b$;
– транзитивным, если $(\forall a,b,c \in A) : ((a,b) \in R, (b,c) \in R) \Rightarrow (a,c) \in R$.

6 Отношение R называется отношением эквивалентности, если оно рефлексивно, симметрично, транзитивно.

7 Бинарное отношение R называется отношением нестрогого порядка, если оно рефлексивно, антисимметрично, транзитивно.

8 Бинарное отношение R называется отношением строгого порядка, если оно антирефлексивно, антисимметрично, транзитивно.

9 Функцией называется бинарное отношение, обладающее свойством однозначности: если $(x, y_1) \in f$ и $(x, y_2) \in f \Rightarrow y_1 = y_2$

10 Область определения функции $f_x = \{x \in X \mid \exists y \in Y : f(x) = y\}$, область значений $f_y = \{y \in Y \mid \exists x \in X : f(x) = y\}$

11 Если $f_x = X$, то функция называется тотальной, в противном случае называется частично определенной.

12 Если $f_y = Y$, то функция f называется сюръективной или сюръекцией.

13 Функция $f : X \to Y$ называется инъекцией, или инъективной, если $(x_1, y) \in f$ и $(x_2, y) \in f \Rightarrow x_1 = x_2$.

14 Функция $f : X \to Y$ одновременно инъективная и сюръективная называется биективной, или биекцией. Биекция – взаимно однозначное соответствие между областью определения и множеством Y.

Пример задачи.

Задание 1
Исходные данные:

A = {1; 3; 7; 8; 15}; B = {21; 35; 40; 60}
Решение:

а) Декартово произведение A×B = {(1; 21); (1; 35); (1; 40); (1; 60); (3; 21); (3; 35); (3; 40); (3; 60); (7; 21); (7; 35); (7; 40); (7; 60); (8; 21); (8; 35); (8; 40); (8; 60); (15; 21); (15; 35); (15; 40); (15; 60)}

б) Выберем те пары, которые соответствуют отношению R = {(1; 21); (1; 35); (1; 40); (1; 60); (3; 21); (3; 60); (7; 21); (7; 35); (8; 40); (15; 60)}

в) Составим матрицу бинарного отношения:

$$\begin{array}{c} \\ 1 \\ 3 \\ 7 \\ 8 \\ 15 \end{array} \begin{array}{c} \begin{array}{cccc} 21 & 35 & 40 & 60 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

г) Составим матрицу обратного отношения:

$$\begin{array}{c} \\ 21 \\ 35 \\ 40 \\ 60 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 3 & 7 & 8 & 15 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

д) Выпишем обратное отношение в виде перечисления пар: R-1 = {(21; 1); (21; 3); (35; 1); (35; 7); (40; 1); (40; 8); (60; 1); (60; 3); (60; 15)}

Задание 2
Исходные данные:

A={2, 5, 15, 23} R={(a, b) (a + b) кратно 5}
Решение:

а) Декартов квадрат A^2 = A×A = {(2; 2); (2; 5); (2; 15); (2; 23); (5; 2); (5; 5); (5; 15); (5; 23); (15; 2); (15; 5); (15; 15); (15; 23); (23; 2); (23; 5); (23; 15); (23; 23)}

б) Выберем те пары, которые соответствуют отношению R = {(2; 23); (5; 5); (5; 15); (15; 5); (15; 15); (23; 2)}

в) Составим матрицу бинарного отношения и обратного отношения:

$$\begin{array}{c} \\ 2 \\ 5 \\ 15 \\ 23 \end{array} \begin{array}{c} \begin{array}{cccc} 2 & 5 & 15 & 23 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ 2 \\ 5 \\ 15 \\ 23 \end{array} \begin{array}{c} \begin{array}{cccc} 2 & 5 & 15 & 23 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

г) Отношение не является рефлексивным, т.к. главная диагональ матрицы не состоит из единиц

д) Матрица бинарного отношения и обратного совпадают, значит, отношение симметрично

е) Отношение не является транзитивным, т.к. если $(a + b)$ и $(b + c)$ кратны 5, $(a + c)$ не обязательно кратно 5 (например, 2 + 23, 23 + 7, 2 + 7 не кратно 5)

ж) Значит, данное отношение не является отношением эквивалентности

Задание 3

Исходные данные:

Q – отношение параллельности на множестве всех прямых плоскости

Решение:

а) Отношение Q рефлексивно, т.к. любая прямая $a \parallel a$.

б) Q симметрично, т.к. если прямая $a \parallel b$, то $b \parallel a$

в) Q транзитивно, т.к. $a \parallel b$, $b \parallel c$, то $a \parallel c$

Задание 4

Исходные данные:

W – отношение предшествования букв в русском алфавите на множест- ве {а; б; в; г; д; е; ж; з}

Решение:

Отношение W антирефлексивно, т.к. любая буква не предшествует самой себе. W антисимметрично, т.к. если буква a предшествует *б*, то *б* не предшествует a. W транзитивно, т.к. если a предшествует *б*, *б* предшествует *в*, то a предшествует *в*.

Значит, отношение W является отношение строгого порядка.

Задание 5

Исходные данные:

{(1, 2), (3, 4), (2, 3), (4, 5)} f_x={1;2;3;4}, f_y={1;2;3;4;5}

Решение:

Это отношение – тотальная функция, т.к. каждому x соответствует единственный y, область определения отношения совпадает с областью определения функции.

Отношение – инъекция, каждому y соответствует единственный x.

Отношение не является сюръекцией, т.к. область значений отношения не совпадает с f_y, не биекция.

Контрольные вопросы:

1 Что такое бинарное отношение?
2 Как можно представить бинарное отношение?
3 Что такое матрица бинарного отношения?

4 Область определения бинарного отношения
5 Область значений бинарного отношения
6 Обратное бинарное отношение
7 Как получить матрицу обратного бинарного отношения?
8 В чем заключается свойство рефлексивности, антирефлексивности би-нарного отношения?
9 В чем заключается свойство симметричности, антисимметричности бинарного отношения?
10 В чем заключается свойство транзитивности бинарного отношения?
11 Эквивалентное бинарное отношение

12 Отношение строгого порядка
13 Отношение нестрогого порядка
14 Что такое функция?
15 Область определения, область значений функции
16 Что такое тотальная функция?
17 Что такое сюръекция?
18 Что такое инъективная функция?
19 Что такое биекция?

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, выполнившему 4 задания или 5 с недочетами.

Оценка «3» ставится студенту, выполнившему 2-3 заданий.

Оценка «2» ставится студенту, выполнившему менее 2-х заданий

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 7

ТЕМА: Формализация предложений с помощью логики предикатов.
ЦЕЛЬ: Научиться формализовывать предложения, используя предикаты и кванторы.

Задание 1.
Записать высказывание и определит его истинность, считая, что все переменные принадлежат множеству действительных чисел.

I вариант	II вариант
А) $(\exists x)\ (\forall y)$: $(x + y = 10)$ $(\forall x)\ (\exists y)\ (\exists z)$: $x*y = z$	А) $(\forall x)\ (\exists y)$: $(x + y = 8)$ $(\forall x)\ (\forall y)$: $(x>y)$
Б) $(\forall x)\ (\exists y)\ (x – y = 7)$ $(\forall x)\ (\forall y)$: $(x+y>0)$	Б) $(\exists x)\ (\forall y)\ (x – y = 5)$ $(\forall z)\ (\exists y)\ (\exists x)$: $x + y = z$

Задание 2.
Записать предложенное высказывание в символичной форме, введя предикаты

I вариант	II вариант
А) У каждого человека есть мать. Некоторые студенты – второкурсники.	А) Существуют города, которые больше Москвы. На каждом доме есть номер.
Б) Каждое материальное тело имеет массу. Существуют кустарники, которые больше чем деревья.	Б) Некоторые космические тела являются астероидами. У любой группы есть классный руководитель

Теоретическая часть

Предикатом называется предложение, содержащее одну или несколько переменных, при подстановке в которые конкретных значений, предложение обращается в высказывание.

Множество М, на котором определен предикат Р(х), называется областью **определения предиката**.

Множество всех элементов $x \in M$, при которых предикат принимает значение «истина», называется **множеством истинности предиката (Т).**

Кроме логических операций над предикатами также определены две кванторные опреации: квантор общности и квантор существования.

Квантор общности (универсальный квантор) - $\forall x$.

$\forall x P(x)$ – для всех (любого) х истинно Р(х). Это высказывание истинно тогда и только тогда, когда предикат Р(х) выполняется для каждого значения переменного х.

Квантор существования - $\exists x$.

$\exists x P(x)$ – существует х, такой что истинно Р(х). Это высказывание истинно тогда и только тогда, когда для некоторых значениях х выполняется предикат Р(х).

Пример 1. Запишите высказывание для символичной записи $(\exists x)(\exists y)$: $(x^2+ y^2 > 25)$.

Определите истинность высказывания, считая, что все переменные принадлежат множеству действительных чисел.

Решение

Данную запись можно представить высказыванием: существует х и существует у, такие что $x^2+ y^2 > 25$. Высказывание является истинным, т.к. можно найти пару чисел х и у, для которых будет выполняться выражение $x^2+ y^2 > 25$ (например, $x = 3$ и $y = 5$).

Пример 2. Запишите высказывание «На каждой улице будет праздник» в символичной форме, введя предикаты.

Решение

1. Найдем область определения
   M: x – множество всех улиц
   y – множество всех праздников
2. Введем предикат P(x, y): x имеет свой Y.
3. Данное высказывание в симвоичной форме запишется в виде: $(\forall x)(\exists y)P(x, y)$

Контрольные вопросы:

1. При каких условиях высказывания $\forall xP(x)$ и $\exists xP(x)$ истинны?
2. Где используются предикаты и кванторы?

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 8

ТЕМА: Операции над предикатами.
ЦЕЛЬ: научиться выполнять логические операции над предикатами.

Задание 1.
Найти множества истинности данных предикатов, если их область определения множество всех действительных чисел.

I вариант	**II вариант**
А) **P (x):** $x^2 - 4 = 0$; Б) **Q(x):** $3x - 2 < 17$	А) **P (x):** $2x^2 - 18 = 0$; Б) **Q(x):** $2x + 3 < 15$
III вариант	**IV вариант**
А) **P (x):** $3x^2 - 12 = 0$; Б) **Q(x):** $5x - 4 > 29$	А) **P (x):** $x^2 - 9 = 0$; Б) **Q(x):** $4x + 6 > 12$

Задание 2.
На множестве M = {1,2,3, … ,20} заданы предикаты: A (x): «x не делится на 5»; B (x): «x – четное число»; C (x): «x – число простое»; D (x): «x кратно 3». Определить следующие предикаты и найти их множества истинности:

I вариант	**II вариант**
A (x) & B(x); $C\ (x) \vee$ D (x); $\bar{B}$ (x); $A\ (x) \rightarrow$ C (x);	C (x) & B (x); $B\ (x) \vee$ D (x); $\bar{C}$ (x); $C\ (x) \rightarrow$ A (x);
III вариант	**IV вариант**
C (x) & D (x); $B(x) \vee$ C (x); $\bar{A}$ (x); $D\ (x) \rightarrow$ C(x);	B (x) & D (x); $A\ (x) \vee$ B (x); $\bar{D}$ (x); $A\ (x) \rightarrow$ B (x);

Теоретическая часть

Предикатом называется предложение, содержащее одну или несколько переменных, при подстановке в которые конкретных значений, предложение обращается в высказывание.

Множество M, на котором определен предикат P(x), называется областью **определения предиката**.

Множество всех элементов $x \in M$, при которых предикат принимает значение «истина», называется **множеством истинности предиката (T).**

Пример 1. Найти множество истинности предиката P(x): $6x^2 - 24 = 0$, если его область определения множество всех действительных чисел.

Решение

Для нахождения множества истинности предиката определим корни уравнения:

$6x^2 - 24 = 0$

$x^2 = 4$

$x_1 = -2$, $x_2 = 2$.

Ответ: Множество истинности T(P) = {-2, 2}.

Для предикатов определены логические операции: отрицание, дизъюнкция, конъюнкция, импликация, эквиваленция и следование.

Пример 2. На множестве M = {1, 2, 3, … ,20} заданы предикаты: A (x): «x не делится на 4»; B (x): «x – нечетное число»; C (x): «x – число простое»; D (x): «x кратно 5». Определить предикаты A(x) & D(x); $A(x) \vee$ C (x); $\bar{B}$ (x); $B(x) \rightarrow$ D (x) и найти их множества истинности.

Решение

1. Найдем множества истинности для исходных предикатов:

   T(A) = {1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15, 17, 18, 19}

T(B) = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19}
T(C) = {1, 2, 3, 5, 7, 11, 13, 17, 19}
T(D) = {5, 10, 15, 20}

2. A(x) & D(x): «число x не делится на 4 и кратно 5»
   T(A&D) = {5, 15}
3. $A(x) \vee C$ (x): «число x не делится на 4 или простое»
   T(A∨C) = {1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14, 15, 17, 15, 19}
4. $\bar{A}$ (x): «x делится на 4»
   T($\bar{A}$) = {4, 8, 12, 16, 20}
5. $B(x) \rightarrow D$ (x): «если x нечетное число, то оно кратно 5»
   T(B→D) = T ($\bar{B}$∨D)
   T ($\bar{B}$) = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20}
   T ($\bar{B}$∨D) = {2, 4, 5, 6, 8, 10, 12, 14, 15, 16, 18, 20}

Кроме логических операций над предикатами также определены две кванторные опреации: квантор общности и квантор существования.

Квантор общности (универсальный квантор) - ∀x.

∀xP(x) – для всех (любого) x истинно P(x). Это высказывание истинно тогда и только тогда, когда предикат P(x) выполняется для каждого значения переменного x.

Квантор существования - ∃x.

∃xP(x) – существует x, такой что истинно P(x). Это высказывание истинно тогда и только тогда, когда для некоторых значениях x выполняется предикат P(x).

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 9

ТЕМА: Получение приведённой нормальной формы формул логики предикатов.
ЦЕЛЬ: Научится записывать формулы логики предикатов в предваренной нормальной форме (ПНФ).

Задание 1.
Привести формулы к предваренной нормальной форме:

1) $\overline{\exists x(P(x) \to \forall y Q(y))}$;
2) $\overline{P \to \exists x R(x)}$;
3) $\overline{\forall x \exists y (A(x) \leftrightarrow A(y))}$.

Теоретическая часть

Логика предикатов – это расширение возможностей логики высказываний, позволяющее строить высказывания с учетом свойств изучаемых объектов или отношений между ними.

Одноместным предикатом $P(x)$ называется функция переменного x, определенная на множестве M и принимающая значения из множества $\{1, 0\}$.

Множество M, на котором определен предикат $P(x)$, называется ***предметной областью*** или областью определения предиката. Множество всех $x \in M$, при которых $P(x) = 1$, называется ***множеством истинности*** предиката.

Многоместным предикатом называется всякая функция n переменных $Q(x_1, x_{2,} \ldots, x_n)$, определенная на множестве $M = M_1 \times M_2 \times \ldots \times M_n$ (декартово произведение) и принимающая на этом множестве одно из двух значений $\{1, 0\}$.

В общем случае n-местным предикатом $P(x_1, x_2, \ldots, x_n)$ называется функция, аргументы которой являются элементами произвольного множества M, а значения принадлежат множеству $\{1, 0\}$, или $P(x_1, x_2, \ldots, x_n) : M^n \to \{1, 0\}$. Элементы множества M называются ***предметными переменными***. Количество предметных переменных есть порядок (местность) предиката.

Чтобы сделать более прозрачной структуру сложных высказываний, удобно ввести специальные обозначения для некоторых часто встречающихся выражений - ***кванторы***. Для их обозначения используются символы:

$\forall$ - квантор всеобщности;

$\exists$ - квантор существования.

Пусть $P(x)$ – одноместный предикат, определенный на множестве М. Тогда под выражением $\forall x P(x)$ будем понимать высказывание, которое принимает значение истина тогда и только тогда, когда $P(x)$ истинно для каждого элемента x множества M. Это высказывание уже не зависит от x. Переменную x в предикате $P(x)$ называют ***свободной***, а в высказывании $\forall x P(x)$ – ***связанной*** квантором всеобщности.

Аналогично, под выражением $\exists x P(x)$ понимают высказывание, которое является истинным, если найдется хотя бы один элемент x множества M, для которого $P(x)$ истинно, и ложным, если ни одного такого элемента во множестве M нет. Высказывание $\exists x P(x)$ не зависит от x, в нем переменная x связана квантором существования.

Из предикатных символов с помощью знаков логических операций и кванторов строятся формулы логики предикатов, которые используются в информационных задачах для описания предметной области. При этом определяется содержание множества предметных переменных M, а каждому предикатному символу придается смысл – задается свойство, которое описывает этот предикат. Таким образом, формулам придается некоторая интерпретация. Одна и та же формула в разных интерпретациях может иметь разные значения.

Если формула F истинна при любых значениях своих аргументов в некоторой интерпретации, то она называется истинной в данной интерпретации. Формула, истинная в любой интерпретации, называется общезначимой. Две формулы логики предикатов называются ***равносильными***, если они принимают одинаковые истинностные значения при любых значениях переменной в любой интерпретации. Все равносильности логики высказываний (табл. 3) справедливы в логике предикатов. Кроме этого, в логике предикатов есть равносильности, связанные с преобразованиями формул, содержащих кванторы (табл. 4).

Таблица 4. Основные равносильности логики предикатов

№	Формула
1	$\neg(\exists x P(x)) \equiv \forall x(\neg P(x))$
2	$\neg(\forall x P(x)) \equiv \exists x(\neg P(x))$
3	$(\forall x P(x)) \& (\forall x Q(x)) \equiv \forall x(P(x) \& Q(x))$
4	$(\exists x P(x)) \vee (\exists x Q(x)) \equiv \exists x(P(x) \vee Q(x))$
5	$(\forall x P(x)) \vee (\forall x Q(x)) \equiv \forall x \forall y(P(x) \vee Q(y))$
6	$(\exists x P(x)) \& (\exists x Q(x)) \equiv \exists x \exists y(P(x) \& Q(y))$

Специальную форму записи формулы логики предикатов называют предваренной нормальной формой (ПНФ).

Алгоритм получения формулы ***в предваренной нормальной форме***:

1) перейти от символов $\to$ и ~ к символам &, $\vee$, $\neg$;
2) внести все отрицания внутрь формулы, “приклеив” их к предикатным символам;
3) вынести все кванторы в начало формулы.

Пример:
Записать формулу логики предикатов $F = \neg(\exists x(P(x) \to \forall y P(y)))$ в ПНФ.

Решение:
В преобразованиях будем использовать законы логики высказываний (табл. 3) и логики предикатов (табл. 4).

$$F \equiv \neg(\exists x(P(x) \to \forall y Q(y))) \equiv \forall x(\neg(\neg P(x) \vee \forall y Q(y))) \equiv$$
$$\equiv \forall x(P(x) \& \neg(\forall y Q(y))) \equiv \forall x(P(x) \& (\exists y \neg Q(y))) \equiv \forall x \exists y(P(x) \& \neg Q(y)).$$

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 10

ТЕМА: Решение логических задач в алгебре предикатов.
ЦЕЛЬ: научиться решать логические задачи в алгебре предикатов, строить высказывания к высказываниям, содержащим кванторы.

Задание 1.
Постройте отрицание к высказываниям, содержащим кванторы.

I вариант	**II вариант**
Все планеты имеют атмосферу. Некоторые люди ходят в театр.	Некоторые студенты учатся на «отлично». Все птицы улетают зимой в теплые края.
III вариант	**IV вариант**
Некоторые машины красного цвета. Все компьютеры подключены к Интернету.	Все кошки любят молоко. Некоторые приборы исправны.

Задание 2.
Проверьте правильность умозаключений.

I вариант	**II вариант**
a) Все адвокаты богаты. Все богатые едят омаров. Все адвокаты едят омаров. b) Некоторые адвокаты богаты. Некоторые врачи богаты. Некоторые врачи – адвокаты.	a) Некоторые марсиане зеленые. Все елки зеленые. Некоторые марсиане – елки. b) Все мужчины любят мясо. Некоторые учителя – мужчины. Некоторые учителя любят мясо.
III вариант	**IV вариант**
a) Все врачи любят музыку. Все поэты любят музыку. Все врачи – поэты. b) Некоторые врачи умные. Все умные люди поэты. Некоторые врачи – поэты.	a) Все машины дорогие. Велосипед не дорогой. Велосипед – не машина. b) Все мужчины смотрят телевизор. Некоторые слесари – мужчины. Некоторые слесари смотрят телевизор.

Теоретическая часть

Для построения отрицания высказываний, содержащих квантор $\frac{\text{общности } (\forall)}{\text{существования } (\exists)}$, достаточно заменить его на другой квантор $\frac{\text{существования } (\exists)}{\text{общности } (\forall)}$ и взять отрицание выражения, на которое этот квантор был «навешан».

Пример 1. Для данных высказываний построить их отрицание.

1) А: «Все целые числа являются простыми».

Данное высказывание содержит квантор общности (слово «все»), заменим его на квантор существования (слово «некоторые») и добавим отрицание с помощью частицы «не».

$\overline{A}$: «Некоторые целые числа не являются простыми»

2) А: «Некоторые люди любят есть репу»

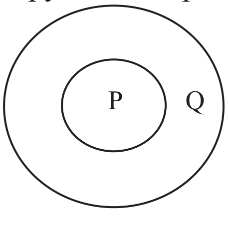
Данное высказывание содержит квантор существования (слово «некоторые»), заменим его на квантор общности («все») и добавим отрицание с помощью частицы «не».

$\overline{A}$: «Все люди не любят есть репу».

Для неформальной проверки правильности умозаключений, включающих утверждения типа «для всех» и «для некоторого», используются диаграммы Эйлера, которые состоят из кругов, изображающих множества.

Утверждению "Все p есть q" соответствует диаграмма, приведенная на рис. 1. На ней круг, изображающий множество p, содержится в круге, изображающем множество q.

Утверждение "Некоторые p есть q" представляется диаграммой на рис. 2. На этой диаграмме пересечение кругов, изображающих множества p и q, непусто.



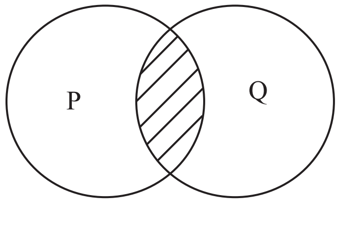
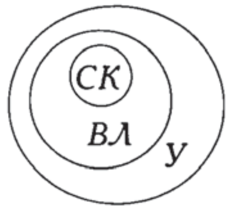


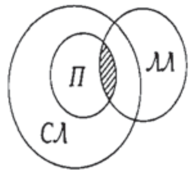
рис. 1 рис. 2

Пример 2. Дано умозаключение. Проверить его правильность.
Все студенты колледжа выдающиеся
Все выдающиеся люди — ученые
Все студенты колледжа — ученые



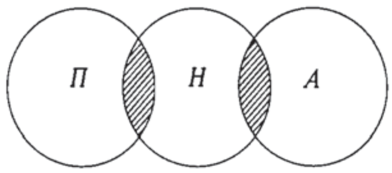
В соответствии с посылками круг, изображающий студентов колледжа (СК), должен быть внутри круга, изображающего выдающихся людей (ВЛ), который, в свою очередь, должен быть внутри круга (У), изображающего ученых. Следовательно, круг студентов колледжа должен находиться внутри круга ученых, и умозаключение является правильным.

Пример 3. Дано умозаключение. Проверить его правильность.
Все поэты счастливы
Некоторые поэты ленивы
Некоторые ленивые люди счастливы



В соответствии с посылками круг, изображающий поэтов (П), должен быть внутри круга, изображающего счастливых людей (СЛ), а пересечение поэтов и ленивых людей (ЛЛ) должно быть непусто. Но это пересечение содержится в круге, изображающем поэтов, так что пересечение ленивых и счастливых людей непусто. Умозаключение правильно.

Пример 4. Дано умозаключение. Проверить его правильность.
Некоторые поэты неудачники
Некоторые атлеты неудачники
Некоторые поэты являются атлетами



Мы видим, что возможно построить такую диаграмму Эйлера, в которой пересечение кругов поэтов (П) и неудачников (Н) непусто и пересечение кругов атлетов (А) и неудачников непусто, так что посылки истинны, но при этом круги поэтов и атлетов не пересекаются, так что следствие не является верным. Следовательно, умозаключение не является правильным.
В основе проверки правильности подобных умозаключений лежит теория силлогистических выводов Аристотеля.

Критерии оценки:
Оценка «5» ставится студенту, полно и правильно выполнившему задание.
Оценка «4» ставится студенту, допустившему 1-2 недочета.
Оценка «3» ставится студенту, допустившему 3-4 недочета.
Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 11

ТЕМА: Программирование машины Поста.
ЦЕЛЬ: Научиться составлять программы для машины Поста.

Задание 1.

Составьте программу для машины Поста.

Вариант	**1**	**2**	**3**	**4**	**5**
задача	1	2	3	4	5
задача	6	7	8	9	10
задача	11	12	13	14	15

1. Составьте программу для машины Поста, которая удаляет каждую третью метку в выражении, если каретка стоит слева от выражения. Слова в выражении разделены пустой ячейкой.
2. Составьте программу для машины Поста, которая увеличивает число в 3 раза, каретка стоит слева от числа
3. Дан массив меток. Проверить, делится ли он нацело на 5. Если да, то слева от него, через одну пустую ячейку поставить метку; если нет, то ничего не делать. Каретка располагается над первой пустой ячейкой, примыкающей к массиву слева..
4. Дан массив меток. Проверить на чётность кол-во меток. Если да, то лента содержит одну метку; если нет, то лента пустая. Каретка располагается где-то над массивом.
5. Дано несколько массивов. Определить их количество. Каретка находится над крайней левой меткой первого массива.
6. Дан массив из 2N-1 меток. Найти и удалить среднюю. Каретка над второй меткой, если считать справа.
7. Дан массив меток. Каретка располагается где-то над массивом, но не над крайними метками. Стереть все метки, кроме крайних
8. Составить программу нахождения разности двух целых неотрицательных чисел a и b. Если a меньше b, то перед разностью через одну пустую ячейку поставить метку. Каретка находится над крайней левой меткой левого числа.
9. Составьте программу для машины Поста, которая увеличивает число в 5 раз, каретка стоит справа от числа
10. Составьте программу для машины Поста, которая удаляет каждую пятую метку в выражении, если каретка стоит справа от выражения. Слова в выражении разделены пустой ячейкой.
11. Дан массив меток. Проверить, делится ли он нацело на 4. Если нет, то слева от него, через одну пустую ячейку поставить метку; если да, то ничего не делать. Каретка располагается над первой пустой ячейкой, примыкающей к массиву слева.
12. Даны два массива разделенные пустой ячейкой, напишите программу которая удалит наименьший массив. Каретка располагается справа от второго массива.
13. Даны два массива разделенные пустой ячейкой, напишите программу которая удалит наибольший массив. Каретка располагается слева от первого массива.
14. Дан массив меток. Каретка располагается справа над первой меткой массива. Стереть все метки, кроме крайних
15. Дан массив меток. Каретка располагается слева над первой меткой массива. Стереть все метки, кроме крайних

Теоретическая часть

1. Устройство машины Поста

Машина Поста - абстрактная машина, которая состоит из ленты и каретки (считывающая и записывающая головка). Лента бесконечна и разделена на секции одинакового размера. В каждую секцию ленты заносится один символ двоичной информации, который подлежит обработке. Один из символов двоичного алфавита - метка "V", другой - пустота. Если в секцию занесена "V" - секция отмеченная, если в секции "V" нет - секция пустая или неотмеченная.

Каретка может передвигаться вдоль ленты влево и вправо. Когда она неподвижна, она стоит против ровно одной секции ленты; говорят, что каретка обозревает эту секцию. А такая секция называется текущей или обозреваемой.

За единицу времени, которая называется шагом, каретка может сдвинуться на одну секцию влево или вправо. Кроме того, каретка может также распознать стоит или нет метка в обозреваемой ею секции, может заносить метку в пустую секцию и может удалять метку из отмеченной ячейки. Команды, по которым каретка должна занести метку в отмеченную секцию или удалить метку из пустой секции являются недопустимыми.

1.2. Система команд машины Поста

Формат команды машины Поста имеет вид: nKm, где:
n - номер текущей команды;
K - команда из системы команд машины Поста (см. табл. 1);
m - ссылка - номер команды, которая будет выполняться следующей.
Последовательность команд из системы команд составляет программу, если:

1. на n - ом месте этой программы будет стоять команда с номером n.
2. ссылке m соответствует реальная команда в программе.

1.3. Таблица машины Поста

В нее непосредственно вписывается алгоритм решения задачи. В столбце "Номер" формируются номера команд алгоритма, начиная с 1, с шагом 1. В столбце "Команды" указываются команды машины Поста. Они выбираются из выпадающего по нажатию клавиши "Enter" или щелчку мыши списка.

В столбце "Отсылка" указываются номера отсылок, т.е. номер строки, которая должна выполняться следующей. Значения в этом столбце не должны быть меньше 1 и больше номера последней команды. В столбце "Комментарии" можно писать различные пояснения, а можно ничего не писать. При выполнении этот столбец игнорируется. Чтобы удалить, вставить, добавить строки, очистить строки или столбцы воспользуйтесь соответствующими пунктами главного или всплывающего меню или панелью инструментов.

Таблица 1. Система команд машины Поста	
a --> b	Сдвиг каретки вправо, содержимое ленты не меняется.
a <-- b	Сдвиг каретки влево, содержимое ленты не меняется.

a V b	В обозреваемую секцию ставится метка "V". Выполнение этой команды возможно только в том случае, если обозреваемая секция пустая, в противном случае команда считается невыполнимой.
a ‡ b	Каретка стирает метку в обозреваемой секции. Выполнение этой команды возможно только в том случае, если обозреваемая секция содержит метку, в противном случае команда считается невыполнимой.
a ? b1 , b2	Команда передачи. Проверяется содержимое текущей секции, если метки нет, то происходит передача управления команде с номером b1, иначе, если метка есть - команде с номером b2. Содержимое ленты не меняется.
a ! [b]	Команда останова машины. Содержимое ленты не меняется. У команды остановки ссылка не обязательна.

2. **Пример.**

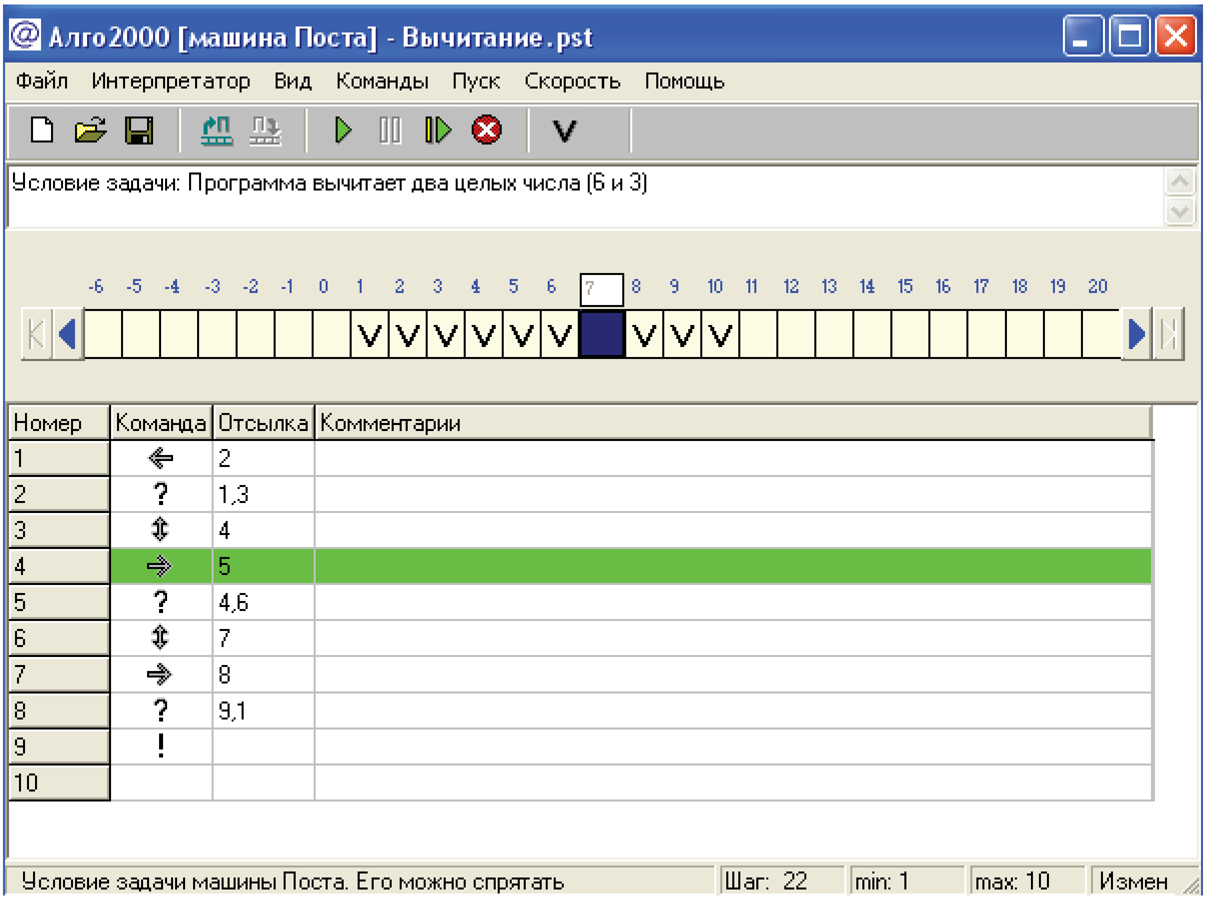
Пример 1:

Выполнить на Машине Поста функцию сложения двух чисел, заданных в позиционной системе счисления. В качестве исходных данных использовать две цифры (№ ПК +10), заданные в позиционной системе счисления. Пример программы сложения двух чисел, заданных в позиционной системе счисления для машины Поста приведен в табл. 2.

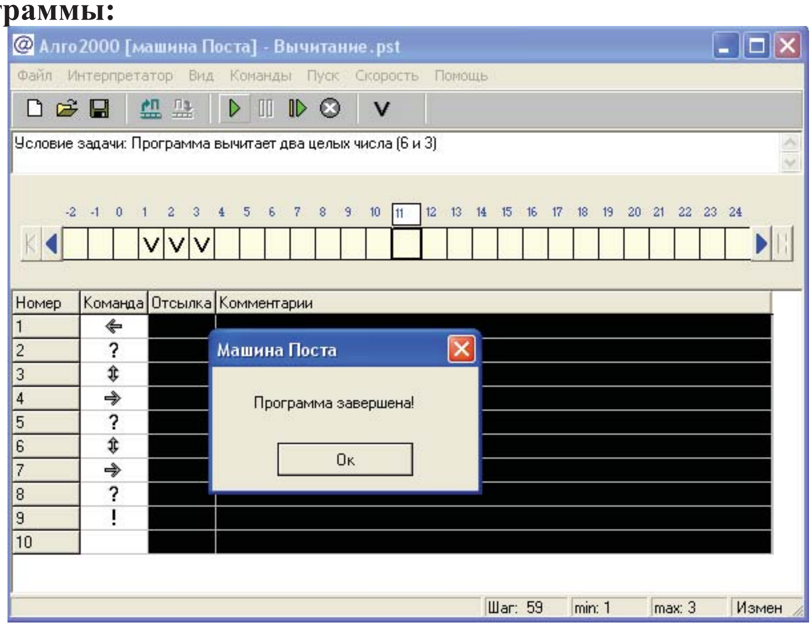
Таблица 2. Пример программы сложения двух чисел для машины Поста				
1	<--	2		Поиск начала первого числа: сдвигаемся влево
2	?	3	1	до тех пор, пока не встретим неотмеченную ячейку.
3	-->	4		Сдвигаемся вправо, на 1-ую метку первого числа и
4	‡	5		удаляем ее
5	-->	6		ищем конец первого числа: сдвигаемся вправо,
6	?	7	5	пока каретка не встанет на неотмеченную ячейку и
7	V	8		ставим метку
8	-->	9		проверяем, заполнился ли промежуток между числами

9	?	1	10	если не заполнился - на первую строку, иначе - переход на
10	!			Конец

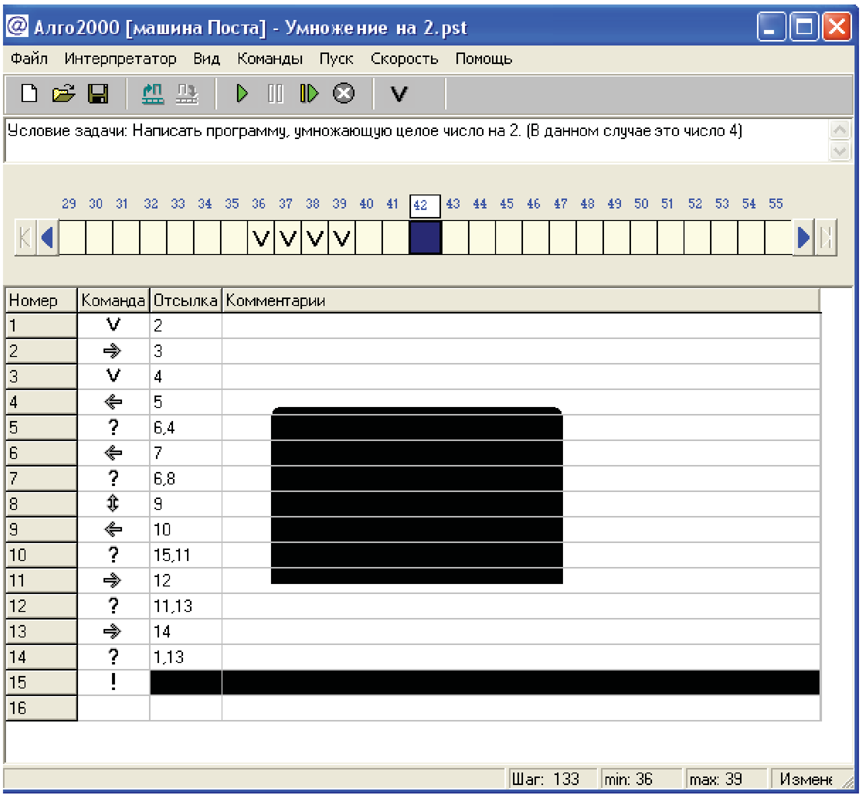
Пример программы вычитания двух целых чисел.



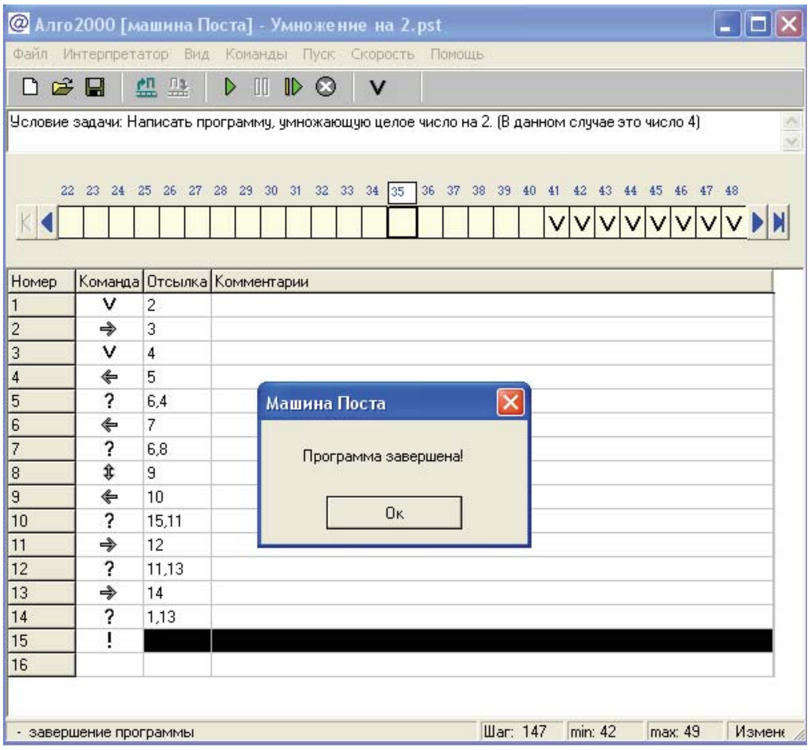
Результат программы:



Пример программы умножения числа на 2.



Результат программы:



3.1. Порядок работы машины Поста

Работа машины Поста состоит в том, что каретка передвигается вдоль ленты и печатает или стирает метки. Чтобы машина Поста работала, надо задать некоторую программу и некоторое состояние машины (т.е. нужно как-то расставить метки по секциям ленты, в частности, можно все секции оставить пустыми, и поставить каретку против одной из секций).

Работа машины на основании заданной программы происходит следующим образом. Машина приводится в начальное состояние и приступает к выполнению первой команды программы. Эта команда выполняется за один шаг, после чего машины приступает к выполнению той команды, номер которой равен отсылке первой команды. Эта команда также выполняется за один шаг, после чего начинается выполнение той команды, номер команды которой равен отсылке предыдущей команды.

Вообще каждая команда выполняется за один шаг, а переход от выполнения одной команды к выполнению другой происходит по следующему правилу: пусть на k-ом шаге выполнялась команда с номером a, тогда:

1) если эта команда имеет единственную отсылку b, то на (k + 1)-ом шаге выполняется команда с номером b;

2) если эта команда имеет две отсылки b1 и b2, то на (k + 1)-ом шаге выполняется одна из двух команд - с номером b1 или с номером b2;

3) если же выполняющаяся на k-ом шаге команда вовсе не имеет отсылки, то на (k +1) - ом шаге и на всех последующих шагах не выполняется никакая команда – машина останавливается.

Возможные случаи останова машины Поста:

1) в ходе выполнения программы машина дойдет до выполнения невыполнимой команды; выполнение программы прекращается, машина останавливается - происходит безрезультатная остановка.

2) в ходе выполнения программы машина дойдет до выполнения команды остановки; программа в этом случае считается выполненной, машина останавливается - происходит результативная остановка.

3) в ходе выполнения программы машина не дойдет до выполнения ни одной из команд, указанных в первых двух вариантах; выполнение программы при этом никогда не прекращается, машина никогда не останавливается - процесс работы машины происходит бесконечно.

3.2. Информационная лента машины Поста

Лента состоит из 1999 ячеек, нумерация от -999 до 999. Ячейка с толстой рамкой, находящаяся в центре ленты - каретка. Всплывающее меню ленты вызывается при щелчке правой кнопкой мыши на ленте. При получении фокуса лентой на ней появляется курсор (синий прямоугольник). Его можно сдвигать по ленте вправо и влево клавишами управления курсором. Удерживая клавишу Shift и нажимая клавиши "Влево"/ "Вправо" можно выделить несколько ячеек. Также ячейки можно выделить мышью: удерживая левую кнопку и выделив нужные. Метки ставятся/ удаляются в ячейках, которые выделены. Это можно сделать несколькими способами (лента должна быть сфокусирована):

• клавишей "Пробел"
• щелкнуть мышью на кнопку "V" на панели инструментов
• выбрать пункт главного или всплывающего меню "поставить/ удалить метки"
• двойным щелчком мыши (в этом случае изменение происходит только в одной ячейке).

Клавиши управления курсором "Вверх" или "Вниз" ставят курсор в каретку.
"Home" - курсор в начало видимой части ленты.
"End" - курсор в конец видимой части ленты.
"Ctrl" + "Влево" - сдвиг каретки влево

"Ctrl" + "Вправо" - сдвиг каретки вправо

Кнопки со стрелкой слева и справа от ленты - прокрутить каретку соответственно влево и вправо. Кнопки со стрелкой и вертикальной палочкой слева и справа от ленты - прокрутить каретку соответственно до крайней левой отмеченной ячейки и до крайней правой отмеченной ячейке. Номера этих ячеек указываются в строке состояния как мин и макс соответственно. Можно непосредственно поставить каретку в нужную ячейку, указав ее номер в редакторе над кареткой. Ленту можно запомнить (пункт меню Команда/ Запомнить ленту) и затем восстановить (Команда/ Восстановить ленту).

3.3. Исполнение алгоритма машины Поста

Запуск алгоритма на исполнение осуществляется:

• пунктом главного меню **Пуск/ Запустить**
• кнопкой **Запустить** на панели инструментов

При этом выполнение программы будет идти до тех пор, пока не встретиться команда стоп или не возникнет какая-нибудь ошибка.

Приостановить выполнение можно:

• пунктом главного меню **Пуск/ Пауза**;
• кнопкой Пауза на панели инструментов.

Если выполнение не было приостановлено, то оно всегда начинается с первой команды алгоритма. Если была нажата пауза, то можно продолжить выполнение с той команды, на которой машина остановилась.

Программу можно выполнить по шагам:

• пунктом главного меню **Пуск/ Пошагово**;
• кнопкой **Пошагово** на панели инструментов.

При этом выполнится очередная команда и выполнение приостановится.

Полностью прервать выполнение программы можно:

• пунктом главного меню **Пуск/ Прервать**;
• кнопкой **Прервать** на панели инструментов.

Регулировать скорость выполнения можно:

• пунктом главного меню **Скорость**

3.4. Ошибки машины Поста

При возникновении этих ошибок происходит прерывание исполнения алгоритма.

• Не указана команда;
• Не указана отсылка;
• Неверно указан номер отсылки;
• Команды с таким номером не существует;
• Нельзя поставить метку, где она уже есть;
• Нельзя стереть метку, где ее нет;
• Ошибка чтения файла;
• Ошибка записи файла.

Критерии оценки:

Оценка «5» ставится студенту, выполнившему 3 задачи.

Оценка «4» ставится студенту, выполнившему 2 задачи или 3 задачи с недочётами.
Оценка «3» ставится студенту, выполнившему 1 задачу или 2 задачи с недочетами.
Оценка «2» ставится студенту, не выполнившему задание

Контроль и оценка осуществляется преподавателем за выполненную работу

Литература:

Кузин А.В., Пескова С.А. Архитектура компьютерных систем. М.: Форум, НИЦ ИНФРА-М, 2011. Стр. 72 – 85

Жмакин А.П. Архитектура ЭВМ. – СПб.: БХВ-Петербург. Стр. 125 – 136

ПРАКТИЧЕСКАЯ РАБОТА 12

ТЕМА: Программирование машины Тьюринга.
ЦЕЛЬ: Научиться составлять программы для машины Тьюринга.

Задание 1.

Составьте программу для машины Тьюринга.

Вариант	1	2	3	4	5
задача	1	2	3	4	5
задача	6	7	8	9	10
задача	11	12	13	14	15

1. Увеличивающую число на 1 (десятичная система счисления, каретка находится на первом символе с левого края числа)
2. Переноса первого символа слова в конец, если алфавит состоит из {a,b,c} и каретка находится на первом символе правого края слова
3. Выполняющую сложение двух чисел от 0 до 99
4. Переноса последнего символа слова в начало, если алфавит состоит из {a,b,c} и каретка находится на первом символе с правого края слова.
5. Которая дублирует крайние символы слова, если минимальная длина слова 3 символа. Алфавит {a,b,c,d}.
6. Которая заменяет символы в слове dd на b, если минимальная длина слова 6 символов и каретка находится в середине слова. Алфавит {a,b,c,d}.
7. Заменить на d каждый второй символ b слове, каретка находится в середине слова. Алфавит {a,b,c,d}.
8. Оставить в слове только первый символ (пустое слово не менять), если алфавит состоит из {a,b,c} и каретка находится на первом символе с правого края слова.
9. Оставить в слове только последний символ (пустое слово не менять), если алфавит состоит из {a,b,c} и каретка находится на первом символе с левого края слова.
10. Если слово чётной длины (0, 2, 4, …), то выдать ответ a, иначе – пустое слово, каретка находится в середине слова. Алфавит {a,b,c,d}.
11. В непустом слове поменять местами его первый и последний символы, если минимальная длина слова 3 символа. Алфавит {a,b}.
12. Определить, является P палиндромом (перевёртышем, симметричным словом) или нет. Ответ: a (да) или пустое слово, если минимальная длина слова 3 символа. Алфавит {a,b}.
13. Уменьшающую число на 1 (десятичная система счисления, каретка находится на первом символе с левого края числа)
14. Переноса последнего символа слова в начало, если алфавит состоит из {a,b,c} и каретка находится на первом символе с правого края слова.
15. Считая непустое слово P записью числа в троичной системе счисления, получить запись этого числа в единичной системе.

Теоретическая часть

Примеры задач

1. Требуется построить машину Тьюринга, которая прибавляет единицу к числу на ленте. Входное слово состоит из цифр целого десятичного числа, записанных в последовательные ячейки на ленте. В начальный момент машина находится против самой правой цифры числа.

∇

*	*	*	*							*	*	*	*	*

Решение. Машина должна прибавить единицу к последней цифре числа. Если последняя цифра равна 9, то ее заменить на 0 и прибавить единицу к предыдущей цифре. Программа для данной машины Тьюринга может выглядеть так:

	*	0	1	2	3	4	5	6	7	8	9
q_1	$1Hq_0$	$1Hq_0$	$2Hq_0$	$3Hq_0$	$4Hq_0$	$5Hq_0$	$6Hq_0$	$7Hq_0$	$8Hq_0$	$9Hq_0$	$0Лq_1$

В этой машине Тьюринга 1 q - состояние изменения цифры, 0 q - состояние останова.

Если в состоянии 1 q автомат видит цифру 0..8, то он заменяет ее на 1..9 соответственно и переходит в состояние q_0 , т.е. машина останавливается. Если же он видит цифру 9, то заменяет ее на 0, сдвигается влево, оставаясь в состоянии q_1 . Так продолжается до тех пор, пока автомат не встретит цифру меньше 9. Если же все цифры были равны 9, то он заменит их нулями, запишет 0 на месте старшей цифры, сдвинется влево и в пустой клетке запишет 1. Затем перейдет в состояние q_0 , т.е. остановится.

2. Дано число *n* в восьмеричной системе счисления. Разработать машину Тьюринга, которая увеличивала бы заданное число *n* на 1. Автомат в состоянии q_1 обозревает некую цифру входного слова.

	*	0	1	2	3	4	5	6	7
q_1	$1Hq_0$	$1Hq_0$	$2Hq_0$	$3Hq_0$	$4Hq_0$	$5Hq_0$	$6Hq_0$	$7Hq_0$	$0Лq_1$

Решение этой задачи аналогично рассмотренному выше примеру.

3. Дана десятичная запись натурального числа *n>1*. Разработать машину Тьюринга, которая уменьшала бы заданное число *n* на 1. Автомат в состоянии q_1 обозревает правую цифру числа. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.

	*	0	1	2	3	4	5	6	7	8	9
q_1		$9Лq_1$	$0Hq_0$	$1Hq_0$	$2Hq_0$	$3Hq_0$	$4Hq_0$	$5Hq_0$	$6Hq_0$	$7Hq_0$	$8Лq_1$

Состояние q_1 - уменьшаем младшую (очередную) цифру на 1. Если она не равна нулю, то после уменьшения сразу - останов, если же младшая цифра равна 0, то вместо нее пишем 9, смещаемся влево и вновь выполняем вычитание. В клетку [a_0 , q_1] машина Тьюринга никогда не попадет, поэтому ее можно не заполнять.

4. Дано натуральное число *n*>1. Разработать машину Тьюринга, которая уменьшала бы заданное число *n* на 1, при этом в выходном слове старшая цифра не должна быть 0. Например, если входным словом было "100", то выходным словом должно быть "99", а не "099". Автомат в состоянии q_1 обозревает правую цифру числа. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.

	q_1	q_2	q_3
*		$*Hq_3$	
0	$9Лq_1$	$0Пq_0$	$*Hq_0$
1	$0Лq_2$	$1Hq_0$	
2	$1Hq_0$	$2Hq_0$	
3	$2Hq_0$	$3Hq_0$	
4	$3Hq_0$	$4Hq_0$	
5	$4Hq_0$	$5Hq_0$	
6	$5Hq_0$	$6Hq_0$	
7	$6Hq_0$	$7Hq_0$	
8	$7Hq_0$	$8Hq_0$	
9	$8Hq_0$	$9Hq_0$	

Состояние q_1 - уменьшаем младшую (очередную) цифру на 1. Если она больше 1, то после уменьшения - сразу останов, если же младшая цифра равна 0, то вместо нее пишем 9, смещаемся влево и вновь выполняем вычитание. Если уменьшаемая цифра равна 1, то вместо нее пишем 0 и переходим в состояние q_2 .

Состояние q_2 - после записи "0" в каком-либо разряде надо проанализировать, не является ли этот ноль старшей незначащей цифрой (т.е. не стоит ли слева от него в записи выходного слова *).

Состояние q_3 - если записанный "0" является старшей незначащей цифрой, то его надо удалить из записи выходного слова.

Те клетки, в которые машина Тьюринга никогда не попадает, оставляем пустыми.

5. На ленте машины Тьюринга находится число, записанное в десятичной системе счисления. Умножить это число на 2. Автомат в состоянии q_1 обозревает крайнюю левую цифру числа.

	q_1	q_2	q_3
*	$*Лq_2$	$*Hq_0$	$1Hq_0$
0	$0Пq_1$	$0Лq_2$	$1Лq_2$
1	$1Пq_1$	$2Лq_2$	$3Лq_2$
2	$2Пq_1$	$4Лq_2$	$5Лq_2$
3	$3Пq_1$	$6Лq_2$	$7Лq_2$
4	$4Пq_1$	$8Лq_2$	$9Лq_2$
5	$5Пq_1$	$0Лq_3$	$1Лq_3$
6	$6Пq_1$	$2Лq_3$	$3Лq_3$
7	$7Пq_1$	$4Лq_3$	$5Лq_3$
8	$8Пq_1$	$6Лq_3$	$7Лq_3$
9	$9Пq_1$	$8Лq_3$	$9Лq_3$

Состояние q_1- поиск правой (младшей) цифры числа.

Состояние q_2 - умножение очередной цифры числа на 2 без прибавления 1 переноса.

Состояние q_3 - умножение очередной цифры числа на 2 с прибавлением 1 переноса.

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

ПРАКТИЧЕСКАЯ РАБОТА 13

ТЕМА: Программирование машины Маркова.
ЦЕЛЬ: Научиться составлять программы для машины Маркова.

Задание 1.

Составьте нормальные алгоритмы Маркова для задач.

Вариант	**1**	**2**	**3**	**4**	**5**
задача	1	2	3	4	5
задача	6	7	8	9	10
задача	11	12	13	14	15

1. Считая непустое слово P записью числа в троичной системе, получить запись этого числа в единичной системе
2. Считая непустое слово P записью двоичного числа, получить это же число, но в четверичной системе. (Замечание: учесть, что в двоичном числе может быть нечётное количество цифр.)
3. A={0,1,2,3}. Считая непустое слово P записью четверичного числа, получить остаток от деления этого числа на 4.
4. В слове P символы «a» расположить слева, а символы «d» справа. Алфавит {a,b,c,d}.
5. Дано число (десятичная система счисления). Определить четное оно или не четное. (Например: 475-нечет, 44-чет)
6. Переведите число из десятичной системы счисления в двоичную систему счисления.
7. Дано число в унарной системе счисления от 0 до 100, переведите его в десятичную систему счисления.
8. Найдите сумму двух чисел от 0 до 99.
9. Найдите разность двух чисел от 0 до 99.
10. A={a,b,c,d}. Дано слово любой длины, содержащее минимум две буквы a. Удалите в слове вторую букву a.
11. A={a,b,c,d}. Дано выражение, состоящее из слов (не более 5). Слова разделены пустой ячейкой. Определите сколько слов в выражении. Например (adc dc bba aad 4)
12. Даны два числа в унарной системе счисления, разделенные пустой ячейкой. Определить какое число больше, если первое число больше, то на ленте оставить только 1, иначе 2. Например: |||||| ||| результат: 1 или || |||| результат: 2.
13. Найдите разность двух чисел от 0 до 99
14. В слове P символы «c» расположить слева, а символы «a» справа. Алфавит {a,b,c,d}
15. Дано число в унарной системе счисления от 0 до 100, переведите его в десятичную систему счисления.

Теоретическая часть

Нормальные алгоритмы Маркова

В разделе рассматриваются задачи на составление нормальных алгоритмов Маркова. Приводится краткое описание этих алгоритмов, на примерах объясняются основные приёмы их составления и предлагаются задачи для самостоятельного решения.

Краткое описание нормальных алгоритмов Маркова

Подстановки

Интересной особенностью нормальных алгоритмов Маркова (НАМ) является то, что в них используется лишь одно элементарное действие – так называемая подстановка, которая определяется следующим образом.

Формулой подстановки называется запись вида α→β (читается «α заменить на β»), где α и β – любые слова (возможно, и пустые). При этом α называется левой частью формулы, а β – правой частью.

Сама подстановка (как действие) задается формулой подстановки и применяется к некоторому слову P. Суть операции сводится к тому, что в слове P отыскивается часть, совпадающая с левой частью этой формулы (т.е. с α), и она заменяется на правую часть формулы (т.е. на β). При этом остальные части слова P (слева и справа от α) не меняются. Получившееся слово R называют ***результатом подстановки***. Условно это можно изобразить так:

$$P\ \boxed{x}\boxed{\alpha}\boxed{y} \quad \rightarrow \quad R\ \boxed{x}\boxed{\beta}\boxed{y}$$

Необходимые уточнения:

1. Если левая часть формулы подстановки входит в слово P, то говорят, что эта формула применима к P. Но если α не входит в P, то формула считается неприменимой к P, и подстановка не выполняется.

2. Если левая часть α входит в P несколько раз, то на правую часть β, по определению, заменяется только первое (самое левое) вхождение α в P:

$$P\ \boxed{x}\boxed{\alpha}\boxed{y}\boxed{\alpha}\boxed{z} \quad \rightarrow \quad R\ \boxed{x}\boxed{\beta}\boxed{y}\boxed{\alpha}\boxed{z}$$

3. Если правая часть формулы подстановки – пустое слово (в таком слове нет ни одного символа), то подстановка α→ сводится к вычеркиванию части α из P (отметим попутно, что в формулах подстановки не принято как-либо обозначать пустое слово):

$$P\ \boxed{x}\boxed{\alpha}\boxed{y} \quad \rightarrow \quad R\ \boxed{x}\boxed{y}$$

4. Если в левой части формулы подстановки указано пустое слово, то подстановка →β сводится, по определению, к приписыванию β слева к слову P:

$$P\ \boxed{x} \quad \rightarrow \quad R\ \boxed{\beta}\boxed{x}$$

Из этого правила вытекает очень важный факт: формула с пустой левой частью применима к любому слову. (Пустое слово всегда присутствует перед любым словом). Отметим также, что формула с пустыми левой и правой частями не меняет слово.

Определение НАМ

Нормальным алгоритмом Маркова (НАМ) называется непустой конечный упорядоченный набор формул подстановки:

$$\begin{cases} \alpha_1 \rightarrow \beta_1 \\ \alpha_2 \rightarrow \beta_2 \\ \quad \ldots \qquad\qquad (k \geq 1) \\ \alpha_k \rightarrow \beta_k \end{cases}$$

В этих формулах могут использоваться два вида стрелок: обычная стрелка (→) и стрелка «с хвостиком» (→). Формула с обычной стрелкой называется обычной формулой, а формула со стрелкой «с хвостиком» – заключительной формулой. Разница между ними объясняется чуть ниже.

Записать алгоритм в виде НАМ – значит предъявить такой набор формул.

Правила выполнения НАМ

К началу выполнения НАМ задается входное слово P. Где именно и кем оно записано – не важно, в НАМ этот вопрос не оговаривается.

Работа НАМ сводится к выполнению последовательности шагов. На каждом шаге входящие в НАМ формулы подстановки просматриваются сверху вниз и выбирается первая из формул, применимых к входному слову P, т.е. самая верхняя из тех, левая часть которых входит в P. Далее выполняется подстановка согласно найденной формуле. Получается новое слово P'.

На следующем шаге это слово P' берется за исходное и к нему применяется та же самая процедура, т.е. формулы снова просматриваются сверху вниз начиная с самой верхней и ищется первая формула, применимая к слову P', после чего выполняется соответствующая подстановка и получается новое слово P". И так далее:

$P \to P' \to P'' \to \ldots$

Следует обратить особое внимание на тот факт, что на каждом шаге формулы в НАМ всегда просматриваются начиная с самой первой.

Необходимые уточнения:

1. Если на очередном шаге была применена обычная формула ($\alpha \to \beta$), то работа НАМ продолжается (переход к следующему шагу).

2. Если же на очередном шаге была применена заключительная формула ($\alpha \to \beta$), то после её применения работа НАМ прекращается. То слово, которое получилось в этот момент, и есть ***выходное слово***, т.е. результат применения НАМ к входному слову.

Как видно, разница между обычной и заключительной формулами подстановки проявляется лишь в том, что после применения обычной формулы работа НАМ продолжается, а после заключительной формулы – прекращается.

3. Если на очередном шаге к текущему слову неприменима ни одна формула, то и в этом случае работа НАМ прекращается, а выходным словом считается текущее слово.

Таким образом, НАМ останавливается по двум причинам: либо была применена заключительная формула, либо ни одна из формул не подошла. То и другое считается «хорошим» окончанием работы НАМ. В обоих случаях говорят, что НАМ примени́м к входному слову.

Однако может случиться и так, что НАМ никогда не остановится; это происходит, когда на каждом шаге есть применимая формула и эта формула обычная. Тогда говорят, что НАМ непримени́м к входному слову. В этом случае ни о каком результате нет и речи.

Примеры на составление НАМ

Рассмотрим примеры, в которых демонстрируются типичные приёмы составления НАМ.

Как и в случае машины Тьюринга, для сокращения формулировки задач будем использовать следующие соглашения:

– буквой P будем обозначать входное слово;

– буквой A будем обозначать алфавит входного слова, т.е. набор тех символов, которые и только которые могут входить во входное слово P (но в процессе выполнения НАМ в обрабатываемых словах могут появляться и другие символы).

Кроме того, в примерах будем справа от формул подстановки указывать их номера. Эти номера не входят в формулы, а нужны для ссылок на формулы при показе пошагового выполнения НАМ.

Пример 1 (удаление и вставка символов) A={a,b,c,d}. В слове P требуется удалить все вхождения символа c, а затем заменить первое вхождение подслова bb на ddd.

Например: abbcabbca $\to$ adddabba

Решение

Прежде всего отметим, что в НАМ, в отличие от машины Тьюринга, легко реализуются вставки и удаления символов. Вставка новых символов в слово – это замена некоторого подслова на подслово с бóльшим числом символов; например, с помощью формулы bb→ddd два символа будут заменены на три символа. При этом не надо заботиться о том, чтобы предварительно освободить место для дополнительных символов, в НАМ слово раздвигается автоматически. Удаление же символов – это замена некоторого подслова на подслово с меньшим числом символов; например, удаление символа с реализуется формулой с→ (с пустой правой частью). При этом никаких пустых позиций внутри слова не появляется, сжатие слова в НАМ происходит автоматически.

С учётом сказанного нашу задачу должен, казалось бы, решать такой НАМ:

$$\begin{cases} c \to & (1) \\ bb \to ddd & (2) \end{cases}$$

Однако это не так. Проверим этот НАМ на входном слове abbcabbca (над стрелками указаны номера применённых формул, а в словах слева от стрелок подчёркнуты для наглядности те части, к которым были применены эти формулы):

$$\mathrm{abb\underline{c}abbca} \xrightarrow{1} \mathrm{abbabb\underline{c}a} \xrightarrow{1} \mathrm{a\underline{bb}abba} \xrightarrow{2} \mathrm{addda\underline{bb}a} \xrightarrow{2} \mathrm{adddaddda}$$

Как видно, НАМ сначала удалил все символы с и только затем заменил первое вхождение bb на ddd. Однако НАМ на этом не остановился и стал заменять остальные вхождения bb. Почему? Дело в том, что, пока применима хотя бы одна формула, НАМ продолжает свою работу. Но нам этого не надо, поэтому мы должны принудительно остановить НАМ после того, как он заменил первое вхождение bb. Вот для этого и нужны заключительные формулы подстановки, после применения которых НАМ останавливается. Следовательно, в нашем алгоритме обычную формулу bb→ddd надо заменить на заключительную формулу bb ↦ ddd:

$$\begin{cases} c \to & (1) \\ bb \mapsto ddd & (2) \end{cases}$$

Вот теперь наш алгоритм будет работать правильно:

$$\mathrm{abb\underline{c}abbca} \xrightarrow{1} \mathrm{abbabb\underline{c}a} \xrightarrow{1} \mathrm{a\underline{bb}abba} \overset{2}{\mapsto} \mathrm{adddabba}$$

Слово, которое получилось после применения заключительной формулы (2), является выходным словом, т.е. результатом применения НАМ к заданному входному слову.

Проверим наш НАМ ещё и на входном слове, в которое не входит bb:

$$\mathrm{d\underline{c}acb} \xrightarrow{1} \mathrm{da\underline{c}b} \xrightarrow{1} \mathrm{dab}$$

К последнему слову (dab) неприменима ни одна формула, поэтому, согласно определению НАМ, алгоритм останавливается и это слово объявляется выходным.

Пример 2 (перестановка символов)

A={a,b}. Преобразовать слово P так, чтобы в его начале оказались все символы a, а в конце – все символы b.

Например: babba → aabbb

Решение

Казалось бы, для решения этой задачи нужен сложный НАМ. Однако это не так, задача решается с помощью НАМ, содержащего всего одну формулу:

{ ba → ab

Пока в слове P справа хотя бы от одного символа b есть символ a, эта формула будет переносить a налево от этого b. Формула перестает работать, когда справа от b нет ни одного a, это и означает, что все a оказались слева от b. Например:

$$\mathrm{\underline{ba}bba} \to \mathrm{abb\underline{ba}} \to \mathrm{ab\underline{ba}b} \to \mathrm{a\underline{ba}bb} \to \mathrm{aabbb}$$

Алгоритм остановился на последнем слове, т.к. к нему уже неприменима наша формула.

Этот и предыдущий примеры показывают, что в НАМ, в отличие от машины Тьюринга, легко реализуются перестановки, вставки и удаления символов. Однако в НАМ возникает другая проблема: как зафиксировать символ (подслово), который должен быть обработан? Рассмотрим эту проблему на следующем примере.

Пример 3 (использование спецзнака)

A={a,b}. Удалить из непустого слова P его первый символ. Пустое слово не менять.

Решение

Ясно, что удалив первый символ слова, надо тут же остановиться. Поэтому, казалось бы, задачу решает следующий НАМ:

$$\begin{cases} a \mapsto & (1) \\ b \mapsto & (2) \end{cases}$$

Однако это неправильный алгоритм, в чём можно убедиться, применив его к слову bbaba:

$$\text{bbaba} \overset{1}{\mapsto} \text{bbba}$$

Как видно, этот НАМ удалил не первый символ слова, а первое вхождение символа a, а это разные вещи. Данный алгоритм будет правильно работать, только если входное слово начинается с символа a. Ясно, что перестановка формул в этом НАМ не поможет, т.к. тогда он будет, напротив, неправильно работать на словах, начинающихся с a.

Что делать? Надо как-то зафиксировать, пометить первый символ слова, например, поставив перед ним какой-либо знак, скажем *, отличный от символов алфавита слова. После этого уже можно с помощью формул вида $*x \mapsto$ заменить этот знак и первый символ x слова на пусто и остановиться:

$$\text{bbaba} \rightarrow \text{*bbaba} \mapsto \text{baba}$$

А как поставить * перед первым символом? Это реализуется формулой $\rightarrow*$ с пустой левой частью, которая, по определению, приписывает свою правую часть слева к слову.

Итого, получаем следующий НАМ:

$$\begin{cases} \quad \rightarrow * & (1) \\ *a \mapsto & (2) \\ *b \mapsto & (3) \end{cases}$$

Проверим его на том же входном слове:

$$\text{bbaba} \overset{1}{\rightarrow} \text{*bbaba} \overset{1}{\rightarrow} \text{**bbaba} \overset{1}{\rightarrow} \text{***bbaba} \overset{1}{\rightarrow} \ldots$$

Как видно, этот алгоритм постоянно приписывает слева звёздочки. Почему? Напомним, что формула подстановки с пустой левой частью применима всегда, поэтому наша формула (1) будет работать бесконечно, блокируя доступ к остальным формулам. Отсюда вытекает очень важное правило: если в НАМ есть формула с пустой левой частью ($\rightarrow\beta$), то её место – только в самом конце НАМ. Учтём это правило и перепишем наш НАМ:

$$\begin{cases} *a \mapsto & (1) \\ *b \mapsto & (2) \\ \quad \rightarrow * & (3) \end{cases}$$

Проверим данный алгоритм:

$$\text{bbaba} \overset{3}{\rightarrow} \text{*bbaba} \overset{2}{\mapsto} \text{baba}$$

Казалось бы, всё в порядке. Однако это не так: наш алгоритм зациклится на пустом входном слове, т.к. постоянно будет применяться формула (3), а согласно условию задачи на таком слове НАМ должен остановиться. В чём причина этой ошибки? Дело в том, что мы ввели знак * для того, чтобы пометить первый символ слова, а затем уничтожить * и этот символ. Но в пустом слове нет ни одного символа, поэтому формулы (1) и (2) ни разу не сработают и постоянно будет выполняться формула (3). Следовательно, чтобы учесть случай пустого входного слова, надо после формул (1) и (2) записать ещё одну формулу, которая уничтожает «одинокую» звёздочку и останавливает алгоритм:

$$\begin{cases} *a \mapsto & (1) \\ *b \mapsto & (2) \\ * \mapsto & (3) \\ \quad \to * & (4) \end{cases}$$

Вот теперь мы, наконец-то, составили правильный алгоритм.

Критерии оценки:

Оценка «5» ставится студенту, полно и правильно выполнившему задание.

Оценка «4» ставится студенту, допустившему 1-2 недочета.

Оценка «3» ставится студенту, допустившему 3-4 недочета.

Оценка «2» ставится студенту, выполнившему задание менее, чем на 50%.

Контроль и оценка осуществляется преподавателем за выполненную работу

СПИСОК ЛИТЕРАТУРЫ

Основные источники:

1. Канцедал С.А. Дискретная математика: Учебное пособие. М.: ИД ФОРУМ, 2015.
2. Спирина М. С., Спирин П.А. Дискретная математика. М.: Академия. 2015.

Дополнительные источники:

1. Иванов, Б.Н. Дискретная математика. Алгоритмы и программы. Расширенный курс М.: Известия, 2015.
2. Куликов В.В. Дискретная математика: учебное пособие. М.: ИД РИОР, 2015.
3. Набебин А.А. Дискретная математика М.: Научный мир, 2015.
4. Просветов Г.И. Дискретная математика: задачи и решения: Учебное пособие М.: БИНОМ. ЛЗ, 2015.
5. Просветов Г.И. Дискретная математика: задачи и решения: учебно-практическое пособие. М.: Альфа-Пресс, 2015.
6. Тюрин С.Ф. Дискретная математика: практическая дискретная математика и математическая логика: учебное пособие М.: ФиС, ИНФРА-М, 2015.

Электронные ресурсы:

1. Библиотека книг по дискретной математике URL: http://www.ph4s.ru/book_pc_diskretka.html (дата обращения 30.09.17)
2. Математический портал «Высшая математика – просто и доступно» URL: http://www.mathprofi.ru/osnovy_matematicheskoj_logiki.html (дата обращения 30.09.17)