

Министерство науки высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Дискретная математика
сборник учебно-методических материалов
для направления подготовки 10.03.01 «Информационная безопасность»

Благовещенск, 2019

*Печатается по решению
редакционно-издательского совета
факультета математики и информатики
Амурского государственного
университета*

Составитель: Акилова И.М.

Дискретная математика: сборник учебно-методических материалов для направления подготовки для направления подготовки 10.03.01 «Информационная безопасность». – Благовещенск: Амурский гос. ун-т, 2019.

© Амурский государственный университет, 2019
© Кафедра Информационных и управляющих систем, 2019
© Акилова И.М., составление

КРАТКОЕ ИЗЛОЖЕНИЕ ЛЕКЦИОННОГО МАТЕРИАЛА

Теория множеств.

Множеством S называется объединение в одно целое объектов, хорошо различимых нашей мыслью или интуицией. Эти объекты называются *элементами* множества S . Такое интуитивное определение дал немецкий математик Г. Кантор. В данном определении важны следующие два момента:

1. Множество- это нечто, состоящее из хорошо различимых объектов.
2. Это нечто мыслится как единое целое.

Множества бывают конечными и бесконечными, Количество элементов в конечном множестве называется **мощностью** множества. Множество, не содержащее ни одного элемента, называется **пустым множеством** и обозначается \emptyset . Множество, включающее в себя все рассматриваемые множества, называется универсальным множеством или **универсумом** и обозначается U . Символом \in обозначается отношение принадлежности. Запись $x \in X$ означает, что элемент x принадлежит множеству X . Если элемент x не принадлежит множеству X , то пишут $x \notin X$.

Множества могут быть заданы следующими способами:

1. перечислением (списком своих элементов);
2. описанием свойств, которыми должны обладать его элементы;
3. порождающей процедурой.

ПРИМЕР

Множество экзаменационных оценок может быть задано:

1. перечислением $A = \{2; 3; 4; 5\}$
2. описанием свойств: $A = \{a \mid a \text{ - экзаменационная оценка}\}$
3. порождающей процедурой: $A = \{a \mid a = 2 + i, i = \overline{0,3}\}$

Подмножеством множества A называется множество B , если любой элемент множества B принадлежит множеству A :

$$A \subseteq B \mid a \in A \Rightarrow a \in B \quad (1)$$

Символом \subseteq обозначается отношение включения. Запись $A \subseteq B$ означает множество A является подмножеством множества B .

Не следует смешивать отношение принадлежности \in и отношение включения \subseteq . Отношение принадлежности применяется к элементам множества, а отношение включения к множествам. Хотя $1 \in \{1\}, \{1\} \in \{\{1\}\}$, не верно, что $1 \in \{\{1\}\}$, так как единственным элементом множества $\{\{1\}\}$ является $\{1\}$.

Если $A \subseteq B$ и $A \neq B$, то $A \subset B$, то есть множество A строго включено в множество B . Символ \subset называется строгим включением.

Свойства подмножеств.

1. **Рефлексивность.** Множество A является подмножеством множества A :

$$A \subseteq A. \quad (2)$$

2. **Транзитивность.** Если множество A является подмножеством множества B , а множество B является подмножеством множества C , то множество A является подмножеством множества C :

$$A \subseteq B, B \subseteq C \Rightarrow A \subseteq C \quad (3)$$

3. **Принцип объемности.** Если множество A является подмножеством множества B , а множество B является подмножеством множества A , то множество A равно множеству B :

$$A \subseteq B, B \subseteq A \Rightarrow A = B \quad (4)$$

Операции над множествами.

1. **Объединением** множеств A и B называется множество, состоящее из всех тех и только тех элементов, которые принадлежат хотя бы одному из множеств A или B :

$$A \cup B = \{x | x \in A \text{ или } x \in B\} \quad (5)$$

2. **Пересечением** множеств A и B называется множество, состоящее из тех и только тех элементов, которые принадлежат множеству A и множеству B :

$$A \cap B = \{x | x \in A \text{ и } x \in B\} \quad (6)$$

3. **Разностью** множества A и B называется множество всех тех элементов, которые принадлежат множеству A и не принадлежат множеству B :

$$A \setminus B = \{x | x \in A \text{ и } x \notin B\} \quad (7)$$

4. **Симметричной разностью** множеств A и B называется множество, состоящее из элементов множества A , не принадлежащих множеству B , и элементов множества B , не принадлежащих множеству A :

$$A \Delta B = \{x | (x \in A \text{ и } x \notin B) \text{ и } (x \in B \text{ и } x \notin A)\} \quad (8)$$

5. **Дополнением** множества A называется множество всех тех элементов, которые не принадлежат множеству A :

$$\bar{A} = \{x | x \notin A\} \quad (9)$$

Для наглядного представления операций над множествами используют **диаграммы Эйлера-Венна**.

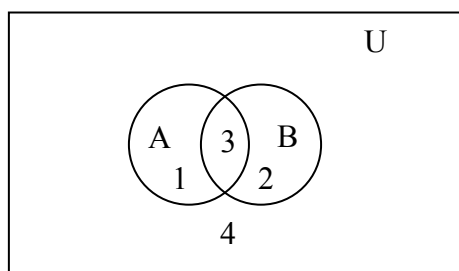


Рис 1. Диаграмма Эйлера-Венна

где $A \cup B$ - это области 1,2,3

$A \cap B$ - это область 3;

$A \setminus B$ - это область 1;

$A \Delta B$ - это область 1,3

\bar{A} - это области 2,4.

Алгебра теории множеств.

Для любых множеств A, B и C выполнимы следующие тождества:

1. Коммутативный закон

$$A \cup B = B \cup A \qquad A \cap B = B \cap A \qquad (9)$$

2. Ассоциативный закон

$$A \cup (B \cup C) = (A \cup B) \cup C \qquad A \cap (B \cap C) = (A \cap B) \cap C \qquad (10)$$

3. Дистрибутивный закон

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \qquad A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \qquad (11)$$

4. Закон поглощения

$$A \cup (A \cap B) = A \qquad A \cap (A \cup B) = A \qquad (12)$$

5. Закон идемпотентности

$$A \cup A = A \qquad A \cap A = A \qquad (13)$$

6. Закон де Моргана

$$\overline{A \cup B} = \overline{A} \cap \overline{B} \qquad \overline{A \cap B} = \overline{A} \cup \overline{B} \qquad (14)$$

7. Закон исключенного третьего

$$A \cup \overline{A} = U \qquad (15)$$

8. Закон противоречия

$$A \cap \overline{A} = \emptyset \qquad (16)$$

9. Операции с универсумом:

$$A \cup U = U \qquad A \cap U = A \qquad (17)$$

10. Операции с пустым множеством:

$$A \cup \emptyset = A \qquad A \cap \emptyset = \emptyset \qquad (18)$$

$$11. \overline{\overline{U}} = \emptyset \qquad \overline{\emptyset} = U \qquad (19)$$

12. Закон двойного дополнения

$$\overline{\overline{A}} = A \qquad (20)$$

$$13. A \setminus B = A \cap \overline{B} \qquad (21)$$

$$14. A \Delta B = (A \setminus B) \cup (B \setminus A) = (A \cap \overline{B}) \cup (B \cap \overline{A}) \qquad (22)$$

При преобразованиях выражений над множествами по законам алгебры логики существуют следующие приоритеты: самой приоритетной операцией является дополнение, затем пересечение и в последнюю очередь объединение.

Решение уравнений алгебры множеств.

Пусть дано уравнение вида:

$$A(X) = B(X) \qquad (23)$$

где X - неизвестное множество. Необходимо определить это неизвестное множество.

Алгоритм решения уравнений алгебры множеств имеет следующий алгоритм:

1. Представляем данное уравнение в следующем виде:

$$A(X) \Delta B(X) = \emptyset \qquad (24)$$

2. Используя алгебру множеств, преобразуем данное уравнение к виду:

$$C \cap X \cup D \cap \overline{X} = \emptyset \qquad (25)$$

где C и D - некоторые множества, не содержащие множество X и его дополнение.

3. Решением уравнения является следующее выражение:

$$D \subseteq X \subseteq \bar{C} \quad (26)$$

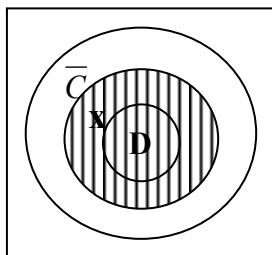


Рис 2. Диаграмма Эйлера-Венна для решения уравнения алгебры множеств.

ПРИМЕР.

Необходимо решить уравнение:

$$X \cup A = B$$

1. Преобразуем данное уравнение:

$$(X \cup A) \Delta B = \emptyset$$

2. С помощью алгебры множеств преобразуем данное выражение следующим образом:

$$(X \cup A) \cap \bar{B} \cup \overline{(X \cup A) \cap B} = X \cap \bar{B} \cup A \cap \bar{B} \cup \bar{X} \cap \bar{A} \cap B = \emptyset$$

В данном выражении присутствует множество $A \cap \bar{B}$, в котором не содержится ни множество X , ни его дополнение, поэтому к этому множеству применяем следующие преобразования:

$$A \cap \bar{B} = A \cap \bar{B} \cap U = A \cap \bar{B} \cap (X \cup \bar{X}) = A \cap \bar{B} \cap X \cup A \cap \bar{B} \cap \bar{X}$$

С учетом данных преобразований имеем:

$$\begin{aligned} X \cap \bar{B} \cup A \cap \bar{B} \cap X \cup A \cap \bar{B} \cap \bar{X} \cup \bar{X} \cap \bar{A} \cap B = \\ = X \cap (\bar{B} \cup A \cap \bar{B}) \cup \bar{X} \cap (A \cap \bar{B} \cup \bar{A} \cap B) = \\ = X \cap \bar{B} \cup \bar{X} \cap (A \Delta B). \end{aligned}$$

Таким образом, имеем множества C и D в следующем виде:

$$\begin{aligned} C &= \bar{B} \\ D &= A \Delta B. \end{aligned}$$

Решением уравнения будет множество:

$$A \Delta B \subseteq X \subseteq B.$$

Решение уравнения (один из вариантов) может быть представлено на диаграмме Эйлера-Венна

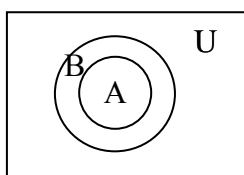


Рис 3 Диаграмма Эйлера-Венна для решения уравнения алгебры множеств.

При изображении решения уравнения алгебры множеств следует иметь в виду, что два множества могут иметь следующие диаграммы Эйлера-Венна

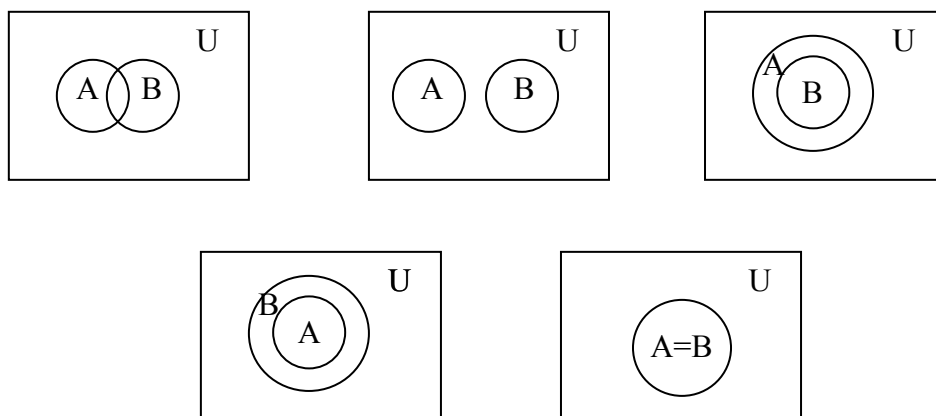


Рис 4 Диаграмма Эйлера-Венна для решения уравнения алгебры множеств.

Кортеж.

Кортеж - это упорядоченный набор элементов. Кортеж характеризуется элементами и их порядком расположения. Элементы кортежа называются **компонентами**. Компоненты нумеруют слева направо. Число компонент определяет длину кортежа. Кортеж обозначается $A = \langle a_1, a_2, \dots, a_n \rangle$.

Кортеж длиной в две компоненты называется парой, кортеж длиной в три компоненты - тройка, длиной в n - n -ка.

Проекцией кортежа на i -тую ось называется его i -тая компонента.

Проекцией кортежа на оси i_1, i_2, \dots, i_q осей называется кортеж, состоящий из i_1, i_2, \dots, i_q компонент, где $2 \leq q \leq n$.

Проекцией кортежа на пустое множество осей является пустой кортеж.

ПРИМЕР

Пусть дан кортеж $A = \langle \alpha, \beta, \chi, \delta \rangle$. Найти проекции на 1 ось, 3 ось, 5 ось, 1 и 4 оси, 4 и 2 оси.

Пр $A_1 = \langle \alpha \rangle$

Пр $A_3 = \langle \chi \rangle$

Пр A_5 не определена

Пр $A_{1,4} = \langle \alpha, \delta \rangle$

Пр $A_{4,2}$ не определена.

Проекция множества.

Проекция множества определена только для множеств, элементами которого являются кортежи одинаковой длины.

Проекцией множества называется множество проекций соответствующих кортежей.

Пример.

$A = \{ \langle 1, 2, 3 \rangle; \langle 4, 5, 6 \rangle; \langle 3, 3, 3 \rangle \}$

Пр $A_1 = \{ \langle 1 \rangle; \langle 4 \rangle; \langle 3 \rangle \}$

Пр $A_{1,3} = \{ \langle 1, 3 \rangle; \langle 4, 6 \rangle; \langle 3, 3 \rangle \}$

Пр $A_{3,1}$ не определена.

График и свойства графика

Графиком называется множество пар. Графики могут задаваться :

1. перечислением: $P = \{ \langle x_1, y_1 \rangle; \langle x_2, y_2 \rangle; \dots \langle x_n, y_n \rangle \}$
2. описанием свойств: $P = \{ \langle x, y \rangle \mid x > y \}$

Пара $\langle a, b \rangle$ называется **инверсией** пары $\langle c, d \rangle$, если $a=d, b=c$.

График P^{-1} называется **инверсией графика P** , если он состоит из инверсий пар графика P .

ПРИМЕР.

$$P = \{ \langle 1, 2 \rangle; \langle 2, 3 \rangle; \langle 3, 4 \rangle; \langle 4, 5 \rangle \}.$$

$$P^{-1} = \{ \langle 2, 1 \rangle; \langle 3, 2 \rangle; \langle 4, 3 \rangle; \langle 5, 4 \rangle \}.$$

График называется **симметричным**, если вместе с каждой парой он содержит её инверсию.

$$Q = P \cap P^{-1} \quad (27)$$

Диагональным называется график вида:

$$\Delta M = \{ \langle x, x \rangle; \langle y, y \rangle; \dots \}, \quad (28)$$

для всех $x, y \in M$.

Композицией графиков $R = P \circ Q$ называется график R , такой что для любой пары $\langle x, y \rangle \in R$ есть такой элемент z , что $\langle x, z \rangle \in P$, а $\langle z, y \rangle \in Q$.

$$R = P \circ Q = \{ \langle x, z \rangle \mid \langle x, y \rangle \in P, \langle y, z \rangle \in Q \}, \quad (29)$$

ПРИМЕР.

1. Пусть заданы графики $P = \{ \langle a, b \rangle; \langle a, c \rangle; \langle f, b \rangle \}$ и $Q = \{ \langle c, c \rangle; \langle b, d \rangle; \langle k, f \rangle; \langle b, m \rangle \}$. Найти композицию графиков P и Q .

$$P \circ Q = \{ \langle a, d \rangle; \langle a, m \rangle; \langle a, c \rangle; \langle f, d \rangle; \langle f, m \rangle \}.$$

2. Пусть заданы графики P и Q :

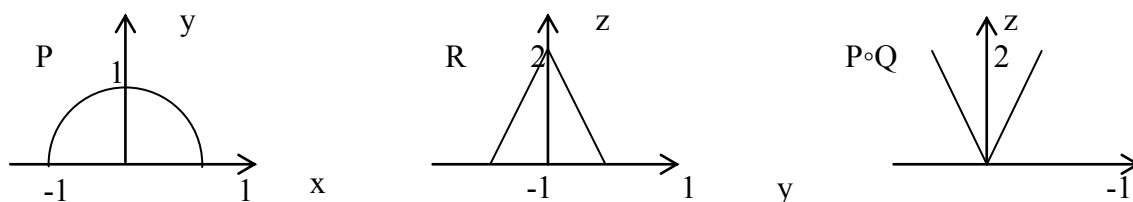


Рис 5. Композиция графиков.

Свойства графиков.

Функциональным графиком называется график, который не содержит пары с одинаковыми первыми и различными вторыми компонентами.

Инъективным графиком называется график, который не содержит пары с одинаковыми вторыми и различными первыми компонентами.

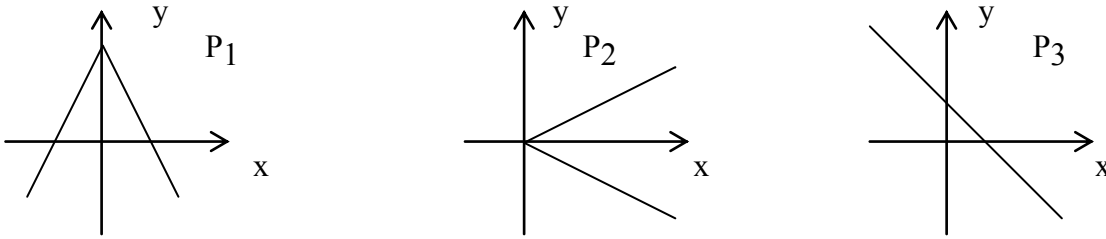


Рис 6. Примеры графиков.

P_1 –График функциональный, но не инъективный.

P_2 –График инъективный, но не функциональный.

P_3 – График функциональный и инъективный.

Возможно другое изображение графиков.. Пусть $X = \{a; b; c; d\}$, а $Y = \{1; 2; 3; 4\}$

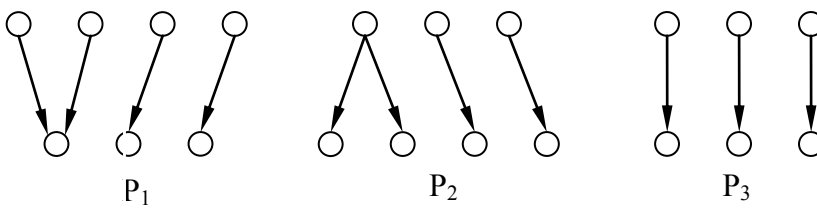


Рис 7 . Примеры графиков.

P_1 –График функциональный, но не инъективный.

P_2 –График инъективный, но не функциональный.

P_3 – График функциональный и инъективный.

Прямое (декартовое) произведение множество.

Прямым (декартовым) произведением множества A и множества B называется множество пар таких, что:

$$A \times B = \{ \langle x, y \rangle \mid x \in A, y \in B \}, \quad (29)$$

ПРИМЕР.

Пусть заданы X и Y : $X = \{x \mid x \leq 2\}$, $Y = \{y \mid y \leq 2\}$. Тогда прямое (декартовое) произведение этих множеств может быть представлено графически: $X \times Y = \{x \mid x \leq 2\}$

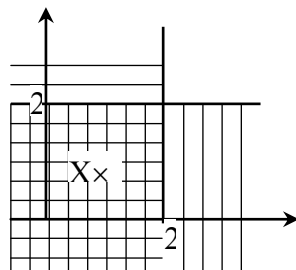


Рис. 8 График прямого произведения множеств X и Y :

Пусть заданы X и Y : $X = \{a; b; c; d; e; f; g; h\}$, $Y = \{1; 2; 3; 4; 5; 6; 7; 8\}$. Тогда прямое (декартовое) произведение этих множеств представляет шахматную доску.

Соответствия.

Соответствием называется тройка вида $\langle P, X, Y \rangle$. При этом X - область отправления (определения), Y - область прибытия (значений), $P \subseteq X \times Y$ - график соответствия.

ПРИМЕР.

Дано соответствие $\langle P, X, Y \rangle$ такое, что:

$$X = \{x \mid x \geq 2\};$$

$$Y = \{y \mid y \geq 5\};$$

$$P = \{\langle x, y \rangle \mid y = x^2\}.$$

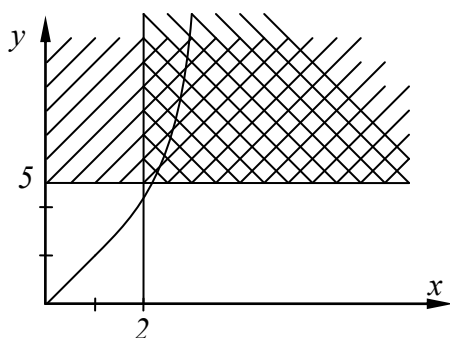


Рис. 9 График соответствия.

Если $P = X \times Y$, то данное соответствие называется **полным**.

Если $P = \emptyset$, то данное соответствие называется **пустым**.

Свойства соответствий.

1. Соответствие называется **функциональным**, если его график функционален.
2. Соответствие называется **инъективным**, если его график инъективен.
3. Соответствие называется **всюду определенным**, если проекция его графика на первую ось совпадает с областью отправления. *Пр* $P_1 = X$.
4. Соответствие называется **сюръективным**, если проекция его графика на вторую ось совпадает с областью прибытия. *Пр* $P_2 = Y$.
5. Соответствие называется **биективным (взаимнооднозначным)**, если оно функционально, инъективно, всюду определено, сюръективно.

ПРИМЕР.

Дано соответствие $\langle P, X, Y \rangle$, где X - множество конфет, Y - множество фантиков, P - "быть упакованным в фантик". Какими свойствами обладает данное соответствие?

Данное соответствие

1. функциональное, так как две и более конфет не может быть упаковано в один фантик;

2. инъективное, так как одна конфета не может быть завернута в два антика одновременно (брак упаковочной техники не рассматривается);
3. не всюду определенное, так как существуют сорта конфет, которые продаются без фантиков (зефир, мармелад);
4. сюръективное, так как фантики без конфет - это мусор;
5. не биективное, так как является несюръективным соответствием.

Отношения.

Отношением называется пара вида $\varphi = \langle \Phi, M \rangle$ такая, что $\Phi \subseteq M^2$ ($M^2 = M \times M$), где Φ - график отношения, M - это множество, между элементами которого существует данное отношение.

ПРИМЕР.

Пусть дано $\varphi = \langle \Phi, M \rangle$, где $M = \{x \mid x \in N\}$, а график отношения определяется как $\Phi = \{ \langle x, y \rangle \mid x > y \}$. Это отношение "X больше Y" на множестве натуральных чисел. Данное отношение задано описанием свойств. Перечислением данное отношение может быть задано следующим образом:

$$\varphi = \langle \Phi, N \rangle,$$

$$\text{где } \Phi = \{ \langle 2, 1 \rangle; \langle 3, 1 \rangle; \langle 3, 2 \rangle; \dots \langle 4, 1 \rangle; \langle 4, 2 \rangle; \dots \}$$

Отношение называется **полным**, если $\Phi = M^2$.

Отношение называется **отношением равенства**, если $\Phi = \Delta M = \{ \langle x, x \rangle; \langle y, y \rangle; \dots \}$.

Отношение называется **отношением неравенства**, если $\Phi = M^2 \setminus \Delta M$.

Запись $x \varphi y$ означает, что между x и y существует отношение φ .

Операции над отношениями.

Пусть даны два отношения $\varphi = \langle \Phi, M \rangle$ и $r = \langle R, M \rangle$. Над данными отношениями могут быть выполнены следующие операции:

1. объединение $\varphi \cup r = \langle \Phi \cup R, M \rangle$.
2. пересечение $\varphi \cap r = \langle \Phi \cap R, M \rangle$.
3. дополнение $\bar{\varphi} = \{ M^2 \setminus \Delta M \}$
4. разность $\varphi \setminus r = \langle \Phi \setminus R, M \rangle$.
5. композиция $\varphi \circ r = \langle \Phi \circ R, M \rangle$.
6. инверсия $\varphi^{-1} = \langle \Phi^{-1}, M \rangle$.

Основные свойства отношений.

1. Рефлексивность.

Отношение называется **рефлексивным**, если для всех x выполняется условие: $x \varphi x$ или $\Delta M \subseteq \Phi$.

2. Анtireфлексивность.

Отношение называется **анtireфлексивным**, если для всех x выполняется условие: $\neg x \varphi x$ (символ " \neg " означает "не выполняется") или $\Delta M \cap \Phi = \emptyset$.

3. Симметричность.

Отношение называется **симметричным**, если для всех x выполняется условие: $x\varphi y \Rightarrow y\varphi x$ или $\Phi = \Phi^{-1}$.

4. Антисимметричность.

Отношение называется **антисимметричным**, если для всех x выполняется условие: $x\varphi y$ и $x \neq y \Rightarrow \neg y\varphi x$ или $\Phi \cap \Phi^{-1} \subseteq \Delta M$.

5. Асимметричность.

Отношение называется **асимметричным**, если для всех x выполняется условие: $x\varphi y \Rightarrow \neg y\varphi x$ или $\Phi \cap \Phi^{-1} = \emptyset$.

6. Связность (полнота).

Отношение называется **связным (полным)**, если для всех x выполняется условие: $x \neq y \Rightarrow x\varphi y$ или $y\varphi x$ или $M^2 \setminus \Delta M \subseteq \Phi \cup \Phi^{-1}$.

7. Транзитивность.

Отношение называется **транзитивным**, если для всех x выполняется условие: $x\varphi y$ и $y\varphi z \Rightarrow x\varphi z$ или $\Phi \circ \Phi \subseteq \Phi$.

ПРИМЕР

Какими свойствами обладает отношение $\varphi = \langle \Phi, X \rangle$, где $X = \{1; 2; a\}$,

$\Phi = \{ \langle 1, 1 \rangle; \langle a, a \rangle; \langle a, 2 \rangle; \langle 2, 2 \rangle \}$.

Определим Φ^{-1} , ΔX :

$\Phi^{-1} = \{ \langle 1, 1 \rangle; \langle a, a \rangle; \langle 2, a \rangle; \langle 2, 2 \rangle \}$

$\Delta X = \{ \langle 1, 1 \rangle; \langle 2, 2 \rangle; \langle a, a \rangle \}$.

Отношение является:

- рефлексивным, так как $\Delta X \subseteq \Phi$;

- антисимметричным, так как $\Phi \cap \Phi^{-1} \subseteq \Delta X$;

- транзитивным, так как $\Phi \circ \Phi = \{ \langle 1, 1 \rangle; \langle a, a \rangle; \langle a, 2 \rangle; \langle 2, 2 \rangle \} \subseteq \Phi$;

- несвязное, так как $X^2 \setminus \Delta X = \{ \langle 1, 2 \rangle; \langle 1, a \rangle; \langle 2, 1 \rangle; \langle 2, a \rangle; \langle a, 1 \rangle; \langle a, 2 \rangle \} \not\subseteq \Phi \cup \Phi^{-1} = \{ \langle 1, 1 \rangle; \langle a, a \rangle; \langle a, 2 \rangle; \langle 2, 2 \rangle; \langle 2, a \rangle \}$.

Отношение называется **отношением эквивалентности**, если оно рефлексивное, симметричное и транзитивное.

Отношение называется **отношением нестрогого (частичного) порядка (\preceq)**, если оно рефлексивное, антисимметричное и транзитивное.

Отношение называется **совершенно нестрого порядка (\preceq)**, если оно рефлексивное, антисимметричное, транзитивное и связное.

Отношение называется **строго порядка ($<$)**, если оно антирефлексивное, антисимметричное и транзитивное.

Отношение называется **совершенно строго порядка ($<$)**, если оно антирефлексивное, транзитивное и связное.

Решетки.

Диаграммы Хассе.

Рассмотрим отношение частичного порядка: "быть подмножеством" на множестве степени $M = \{1, 2\}$.

$\varphi = \langle \Phi, M \rangle$, где

$\Phi = \{ \langle \{\emptyset\}, \{\emptyset\} \rangle; \langle \{\emptyset\}, \{1\} \rangle; \langle \{\emptyset\}, \{2\} \rangle; \langle \{\emptyset\}, \{1,2\} \rangle; \langle \{1\}, \{1\} \rangle; \langle \{1\}, \{1,2\} \rangle; \langle \{2\}, \{2\} \rangle; \langle \{2\}, \{1,2\} \rangle; \langle \{1,2\}, \{1,2\} \rangle \}$.

Графически данное отношение можно изобразить следующим образом:

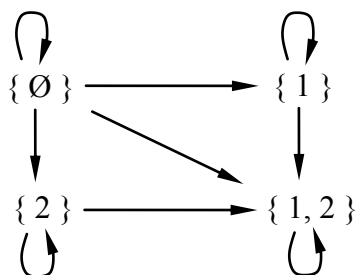


РИС 10 Графическое изображения отношения

Отношение является рефлексивным (графически это отображается петлями), антисимметричным (графически - однонаправленные стрелки), транзитивным (графически - транзитивными замыканиями вида:

Для отношений частичного порядка применимы диаграммы Хассе, которые строятся на основе обыкновенной диаграммы следующим образом:

1. рефлексивные петли и транзитивные связи не изображаются;
2. большие элементы (элементы, в которые входят стрелки) располагают выше.

Таким образом, диаграмма Хассе для вышеприведенного примера будет выглядеть следующим образом:

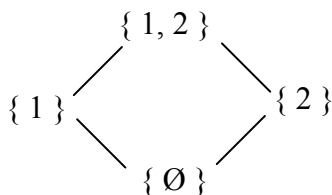


РИС 11 Диаграмма Хассе

Для частично упорядоченного множества справедливо следующее:

1. Элемент $a \in A$ называется **наибольшим (наименьшим)**, если для всех $x \in A$ выполняется $x \leq a$ ($a \leq x$). Если наибольший (наименьший) элемент существует, то он единственный.

2. Элемент $a \in A$ называется **максимальным (минимальным)**, если нет в множестве A элементов, больших (меньших), чем a . Максимальных (минимальных) элементов может быть несколько.

3. Пусть $B \subseteq A$. Элемент $a \in A$ называется **мажорантой (минорантой)**, если для всех $x \in B$ этот элемент является наибольшим (наименьшим).

4. Множество мажорант B образует **верхнюю границу множества B** . Множество минорант B образует **нижнюю границу множества B** .

5. Наименьший элемент верхней границы называется **точной верхней границей**, или **supremum (sup) B** . Наибольший элемент нижней границы называется **точной нижней границей**, или **infimum (inf) B** .

6. Частично упорядоченное множество, у которого для любой пары элементов определен и существует \sup и \inf , называется **решеткой**.

ПРИМЕР

Пусть дано отношение, представленное на диаграмме Хассе

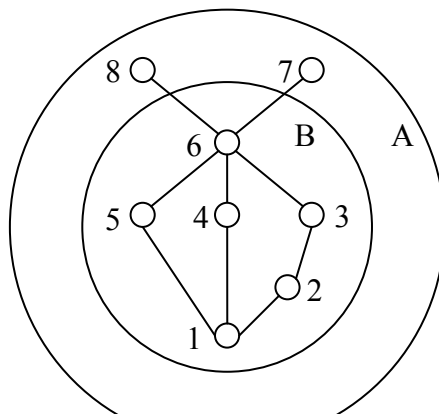


РИС 12 Диаграмма Хассе

Отношение A не является решеткой, т.к. элементы 7 и 8 не имеют sup. Отношение B является решеткой, т.к. любая пара имеет sup и inf.

Алгебраическое представление решеток.

Введем обозначения: $\sup(a,b)=a \cup b$, $\inf(a,b)=a \cap b$. Для решетки справедливы следующие свойства:

1. Коммутативный:

$$a \cup b = b \cup a$$

$$a \cap b = b \cap a$$

2. Ассоциативный:

$$a \cup (b \cap c) = (a \cup b) \cap c$$

$$a \cap (b \cup c) = (a \cap b) \cup c$$

3. Идемпотентности:

$$a \cup a = a$$

$$a \cap a = a$$

4. Поглощения:

$$a \cup (a \cap b) = a$$

$$a \cap (a \cup b) = a$$

Решетки, для которой выполняется дистрибутивный закон:

$$a \cup (b \cap c) = (a \cup b) \cap (a \cup c)$$

$$a \cap (b \cup c) = (a \cap b) \cup (a \cap c)$$

называется **дистрибутивной** решеткой.

Решетка называется **ограниченной**, если она имеет максимальный и минимальный элемент.

ПРИМЕР

Пусть дана решетка (рис. 13). Определить является ли решетка дистрибутивной.

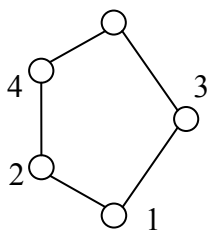


РИС 13 Диаграмма Хассе решетки

Решетка не является дистрибутивной, т.к. для элементов $\{2;3;4\}$ не выполняется дистрибутивный закон:

$$2 \cup (3 \cap 4) = 2 \cup 1 = 2$$

$$(2 \cup 3) \cap (2 \cup 4) = 5 \cap 4 = 4$$

$$2 \cup (3 \cap 4) \neq (2 \cup 3) \cap (2 \cup 4)$$

Дана решетка $\varphi = \langle \Phi, M \rangle$,

где $M = \{x \mid 0 < x < 1\}$, $\Phi = \{ \langle x, y \rangle \mid x < y \}$. Эта решетка не является , так как не определен максимальный элемент (0.9999999999 ...) и минимальный элемент (0.0000000...1).

Обозначим в ограниченной решетке максимальный элемент 1, а минимальный элемент 0. Элемент \bar{a} называется **дополнением элемента** a в данной решетке, если $a \cup \bar{a} = 1$ и $a \cap \bar{a} = 0$. Решетка называется с дополнением, если каждый элемент имеет хотя бы одно дополнение.

ПРИМЕР

Рассмотрим решетку, представленную на рис. 13. Найдем дополнения для каждого элемента решетки

$$\bar{1} = 5$$

$$\bar{2} = 3$$

$$\bar{3} = 2$$

$$\bar{4} = 3$$

$$\bar{5} = 1$$

Данная решетка является решеткой с дополнением.

Ограниченная дистрибутивная решетка с дополнением называется **булевой** решеткой.

На рис . 14 представлены дистрибутивные решетки

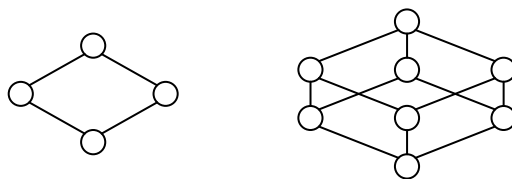


РИС. 14. Примеры булевых решеток

Математическая логика

Высказывания и операции над высказываниями.

Высказыванием называется повествовательное предложение, о котором можно сказать истинно оно или ложно.

1. Москва - столица России.
2. Если студент учится на отлично, то он получит красный диплом.
3. Осадки - это снег или дождь.
4. Курица – не птица.
5. Пейте томатный сок.
6. Я лгу.
7. $23 < 5$

Высказываниями являются 1, 2, 3, 4 и 7 предложения. Предложение 5 не является высказыванием, так как про него нельзя сказать истинно оно или ложно. Предложение 6 является логическим парадоксом.

Элементарным высказыванием называется высказывание, которое содержит одно утверждение (предложения 1, 7).

Сложное (составное) высказывание состоит из элементарных высказываний связанных с помощью следующих предлогов и частиц: *И, НЕ, ИЛИ, ЕСЛИ - ТО, ТОГДА И ТОЛЬКО ТОГДА* и другие. Предложения 2, 3, 4 являются сложными высказываниями.

Операции над высказываниями.

Отрицанием высказывания x называется новое высказывание, которое истинно, если высказывание ложное и наоборот. Таблица истинности операции отрицания имеет вид:

x	\bar{x}
0	1
1	0

Дизъюнкцией двух высказываний x и y (логическое «или») называется новое высказывание, которое будет истинным тогда когда, когда хотя бы одно из высказываний будет истинным.

x	y	$x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Конъюнкцией двух высказываний x и y (логическое «и») называется новое высказывание, которое будет истинным тогда когда, когда оба высказывания истинны. Обозначение операции конъюнкция - $\&(\cdot)$

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

Импликацией двух высказываний x и y («если – то») называется новое высказывание, которое ложно тогда, когда x (*предпосылка*) – истинно, а y (*следствие*) – ложно.

x	y	$x \rightarrow y$
0	0	1
0	1	1
1	0	0
1	1	1

Эквивалентностью двух высказываний x и y («тогда и только тогда») называется новое высказывание, которое будет истинно, если высказывания x и y будут одновременно истинны или ложны.

x	y	$x \leftrightarrow y$
0	0	1
0	1	0
1	0	0
1	1	1

Неодназначностью (суммой по модулю два) двух высказываний x и y («тогда и только тогда») называется новое высказывание, которое будет истинно тогда когда одно из высказываний x или y истинно, а другое ложно.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Штрих Шеффера (логическое «и - не») высказываний x и y - это новое высказывание, которое будет ложно тогда и только тогда когда оба высказывания истинны.

x	y	$x y$
0	0	1
0	1	1
1	0	1
1	1	0

Стрелка Пирса (логическое «или - не») высказываний x и y - это новое высказывание, которое будет истинно тогда и только тогда когда оба высказывания ложны.

x	y	$x y$
0	0	1
0	1	0
1	0	0
1	1	0

Для операций справедливы следующие приоритеты: \neg , $\&$, \vee , \rightarrow , \leftrightarrow .

Формулы математической логики.

Формулой математической логики называется сложное высказывание, которое получено из элементарных высказываний с использованием логических операций.

Две формулы **равносильны**, если они принимают одинаковые логические значения на любом наборе значений входящих в формулу элементарных высказываний. Равносильность формул обозначается – $A \equiv B$.

Формулы равносильности.

1) Коммутативность

$$A \vee B \equiv B \vee A$$

$$A \& B \equiv B \& A$$

2) Ассоциативность

$$A \vee (B \vee C) \equiv (A \vee B) \vee C$$

$$A \& (B \& C) \equiv (A \& B) \& C$$

3) Дистрибутивность

$$A \vee (B \& C) \equiv (A \vee B) \& (A \vee C)$$

$$A \& (B \vee C) \equiv (A \& B) \vee (A \& C)$$

4) Идемпотентность

$$A \vee A \equiv A$$

$$A \& A \equiv A$$

5) Поглощение

$$A \vee (A \& B) \equiv A$$

$$A \& (A \vee B) \equiv A$$

6) Закон де Моргана

$$\overline{A \vee B} \equiv \bar{A} \& \bar{B}$$

$$\overline{A \& B} \equiv \bar{A} \vee \bar{B}$$

7) Закон исключающий третьего

$$A \vee 1 \equiv 1$$

$$A \& 1 \equiv A$$

8) Закон противоречия

$$A \vee \emptyset \equiv A$$

$$A \& \emptyset \equiv \emptyset$$

9) Закон двойного отрицания

$$\overline{\overline{A}} \equiv A$$

$$10) \quad \overline{0} \equiv 1, \quad \overline{1} \equiv 0$$

$$11) \quad A \rightarrow B \equiv \overline{A} \vee B$$

$$12) \quad A \leftrightarrow B \equiv (A \rightarrow B) \& (B \rightarrow A)$$

$$13) \quad A \oplus B \equiv A \& \overline{B} \vee \overline{A} \& B$$

$$14) \quad A | B \equiv \overline{A \& B} \equiv \overline{A} \vee \overline{B}$$

$$15) \quad A \downarrow B \equiv \overline{A \vee B} \equiv \overline{A} \& \overline{B}$$

ПРИМЕР

Доказать: $(x \vee y) \cdot (x \vee \overline{y}) \equiv x$

$$(x \vee y) \cdot (x \vee \overline{y}) \equiv xx \vee x\overline{y} \vee yx \vee y\overline{y} \equiv x \vee x\overline{y} \vee xy \vee 0 \equiv x \vee x\overline{y} \vee xy \equiv x \vee xy \equiv x$$

Представление произвольной функции алгебры логики в виде формулы алгебры логики

Пусть $F(x_1, x_2, \dots, x_n)$ - произвольная функция алгебры логики n переменных.

Рассмотрим формулу

$$F(1,1,\dots,1,1)x_1 x_2 \dots x_{n-1} x_n \vee F(1,1,\dots,1,0)x_1 x_2 \dots x_{n-1} \overline{x_n} \vee \dots \vee F(0,0,\dots,0,0)\overline{x_1} \overline{x_2} \dots \overline{x_{n-1}} \overline{x_n} \quad (2.1)$$

которая составлена следующим образом: каждое слагаемое этой логической суммы представляет собой конъюнкцию, в которой первый член является значением функции F при некоторых определенных значениях переменных x_1, x_2, \dots, x_n , остальные же члены конъюнкции представляют собой переменные или их отрицания. При этом под знаком отрицания находятся те и только те переменные, которые в первом члене конъюнкции имеют значение 0..

Вместе с тем формула (2.1) содержит в виде логических слагаемых всевозможные конъюнкции указанного вида.

Ясно, что формула (2.1) полностью определяет функцию $F(x_1, x_2, \dots, x_n)$. Иначе говоря, значения функции F и формулы (2.1) совпадают на всех наборах значений переменных x_1, x_2, \dots, x_n . То есть функция

Составление формул по таблице истинности. $F(x_1, x_2, \dots, x_n)$ может быть представлена в виде:

$$F(x_1, x_2, \dots, x_n) = F(1,1,\dots,1,1)x_1 x_2 \dots x_{n-1} x_n \vee F(1,1,\dots,1,0)x_1 x_2 \dots x_{n-1} \overline{x_n} \vee F(1,1,\dots,0,1)x_1 x_2 \dots \overline{x_{n-1}} x_n \vee \dots \vee F(0,0,\dots,0,0)\overline{x_1} \overline{x_2} \dots \overline{x_{n-1}} \overline{x_n} \quad (2.2)$$

ПРИМЕР Пусть функция $F(x_1, x_2, x_3)$ имеет следующую таблицу истинности:

x_1	x_2	x_3	$F(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Тогда функция $F(x_1, x_2, x_3)$ может быть определена в следующем виде:

$$\begin{aligned}
 F(x_1, x_2, x_3) &= 1 \cdot \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \vee 0 \cdot \overline{x_1} \cdot \overline{x_2} \cdot x_3 \vee 0 \cdot \overline{x_1} \cdot x_2 \cdot \overline{x_3} \vee \\
 &\vee 1 \cdot \overline{x_1} \cdot x_2 \cdot x_3 \vee 0 \cdot x_1 \cdot \overline{x_2} \cdot \overline{x_3} \vee 1 \cdot x_1 \cdot \overline{x_2} \cdot x_3 \vee 1 \cdot x_1 \cdot x_2 \cdot \overline{x_3} \vee \\
 &\vee 0 \cdot x_1 \cdot x_2 \cdot x_3 \equiv 1 \cdot \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \vee 1 \cdot \overline{x_1} \cdot x_2 \cdot x_3 \vee 1 \cdot x_1 \cdot \overline{x_2} \cdot x_3 \vee 1 \cdot x_1 \cdot x_2 \cdot \overline{x_3} \equiv \\
 &\equiv \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \vee x_1 \cdot x_2 \cdot x_3 \vee x_1 \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot x_2 \cdot \overline{x_3}
 \end{aligned}$$

Нетрудно заметить, что для определения функции берутся только те наборы переменных x_1, x_2, x_3 , при которых функция принимает значения 1, что значительно упрощает процедуру определения функции F .

Формула (2.1) обладает свойствами:

1. Каждое логическое слагаемое формулы содержит все переменные, входящие в функцию $F(x_1, x_2, \dots, x_n)$.
2. Все логические слагаемые формулы различны.
3. Ни одно логическое слагаемое формулы не содержит одновременно переменную и ее отрицание.
4. Ни одно логическое слагаемое формулы не содержит одну и ту же переменную дважды.
5. Перечисленные свойства называются **свойствами совершенства**.

Различные формы представления высказываний

Литерой – называется элемент высказывания x или её отрицание.

Элементарной дизъюнкцией называется выражение следующего вида:

$$L_1 \vee L_2 \vee \dots \vee L_n, \quad (2.2)$$

где L_1 – литера.

Элементарной конъюнкцией называется выражение следующего вида:

$$L_1 \cdot L_2 \cdot \dots \cdot L_n, \quad (2.3)$$

Дизъюнктивной нормальной формой (ДНФ) формулы A называется выражение вида:

$$КНФ A \equiv K_1 \vee K_2 \vee \dots \vee K_n, \quad (2.4)$$

где K_1 – элементарная конъюнкция.

Конъюнктивной нормальной формой (КНФ) формулы A называется выражение вида:

$$\text{ДНФ } A \equiv D_1 \cdot D_2 \cdot \dots \cdot D_n, \quad (2.5)$$

где D_1 – элементарная дизъюнкция.

Любую формулу можно представить в виде ДНФ или КНФ.

ПРИМЕР

Пусть дана формула

$$A = \overline{x \vee y} \leftrightarrow x \cdot y$$

Требуется получить ДНФ и КНФ данной формулы.

Применяя формулы равносильности, получаем КНФ A :

$$\begin{aligned} A = \overline{x \vee y} \leftrightarrow xy &\equiv \overline{(x \vee y)} \cdot \overline{x \cdot y} \vee (x \vee y) \cdot x \cdot y \equiv (x \vee y) \cdot (\overline{x} \vee \overline{y}) \vee \overline{x} \cdot \overline{y} \cdot x \cdot y \equiv \\ &\equiv (x \vee y) \cdot (\overline{x} \vee \overline{y}) \end{aligned}$$

Применяя формулы равносильности, получаем ДНФ A :

$$\begin{aligned} A = \overline{x \vee y} \leftrightarrow xy &\equiv \overline{(x \vee y)} \cdot \overline{x \cdot y} \vee (x \vee y) \cdot x \cdot y \equiv (x \vee y) \cdot (\overline{x} \vee \overline{y}) \vee \overline{x} \cdot \overline{y} \cdot x \cdot y \equiv \\ &\equiv (x \vee y) \cdot (\overline{x} \vee \overline{y}) \equiv x \cdot \overline{x} \vee x \cdot \overline{y} \vee \overline{x} \cdot y \vee \overline{x} \cdot \overline{y} \equiv x \cdot \overline{y} \vee \overline{x} \cdot y \end{aligned}$$

Совершенной дизъюнктивной нормальной формой (СДНФ) формулы A называется такая ДНФ, для которой выполняются следующие условия:

1. Все элементарные конъюнкции, входящие в ДНФ A , различны.
2. Все элементарные конъюнкции, входящие в ДНФ A , содержат литеры, соответствующие всем переменным.
3. Каждая элементарная конъюнкция, входящая в ДНФ A , не содержит двух одинаковых литер.
4. Каждая элементарная конъюнкция, входящая в ДНФ A , не содержит переменную и ее отрицание.

СДНФ A можно получить двумя способами:

1. по таблице истинности;
2. с помощью равносильных преобразований.

Первый способ получения СДНФ A рассмотрен выше. Рассмотрим второй способ, который состоит в следующем:

С помощью равносильных преобразований формулы A получают ДНФ A . При этом в полученной ДНФ возможны следующие ситуации:

1. Элементарная конъюнкция B ДНФ A не содержит переменную x_i , тогда используются следующие равносильные преобразования:

$$B \equiv B1 \equiv B(x_i \vee \overline{x_i}) \equiv Bx_i \vee B\overline{x_i}$$

2. Если в ДНФ A входят две одинаковые элементарные конъюнкции, то используя следующее равносильное преобразование:

$$B \vee B \equiv B,$$

одну элементарную конъюнкцию можно отбросить.

3. Если элементарная конъюнкция B ДНФ A содержит одновременно переменную x_i и ее отрицание, то используя следующие равносильные преобразования:

$$Bx_i \bar{x}_i \equiv B0 = 0,$$

эту элементарную конъюнкцию можно отбросить

4. Если элементарная конъюнкция B ДНФ A содержит дважды переменную x_i , то используя следующее равносильное преобразование:

$$Bx_i x_i \equiv Bx_i,$$

одну переменную x_i можно отбросить

СДНФ формулы A существует в единственном виде.

ПРИМЕР

Получить СДНФ формулы $A = x \rightarrow yz$

С помощью равносильных преобразований получаем СДНФ A :

$$\begin{aligned} A = x \rightarrow yz &\equiv \bar{x} \vee yz \equiv \bar{x}(y \vee \bar{y})(z \vee \bar{z}) \vee yz(x \vee \bar{x}) \equiv \bar{x}yz \vee \bar{x}\bar{y}z \vee \bar{x}y\bar{z} \vee \bar{x}y\bar{z} \vee \bar{x}yz \vee \\ &\vee \bar{x}yz \equiv \bar{x}yz \vee \bar{x}\bar{y}z \vee \bar{x}y\bar{z} \vee \bar{x}yz \vee \bar{x}yz \end{aligned}$$

С помощью таблицы истинности получаем СДНФ A :

x	y	z	\bar{z}	$y\bar{z}$	$A = x \rightarrow yz$
0	0	0	1	0	1
0	0	1	0	0	1
0	1	0	1	1	1
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0

$$\text{СДНФ } A = \bar{x}\bar{y}z \vee \bar{x}y\bar{z} \vee \bar{x}yz \vee \bar{x}yz \vee \bar{x}yz$$

Очевидно, что в результате двух способов совпадает.

Совершенной конъюнктивной нормальной формой (СКНФ) формулы A называется такая КНФ, для которой выполняются следующие условия:

1. Все элементарные дизъюнкции, входящие в КНФ A , различны.
2. Все элементарные дизъюнкции, входящие в КНФ A , содержат литеры, соответствующие всем переменным.
3. Каждая элементарная дизъюнкция, входящая в КНФ A , не содержит двух одинаковых литер.
4. Каждая элементарная дизъюнкция, входящая в КНФ A , не содержит переменную и ее отрицание.

СКНФ A можно получить двумя способами:

1. по таблице истинности;
2. с помощью равносильных преобразований.

По первому способу по таблице истинности получаем СДНФ \bar{A} , а СКНФ A можно получить, следуя следующему правилу

$$СКНФ A = \overline{СДНФ \bar{A}}$$

С помощью равносильных преобразований формулы A получают КНФ A . При этом в полученной КНФ возможны следующие ситуации:

1. Элементарная дизъюнкция B КНФ A не содержит переменную x_i , тогда используются следующие равносильные преобразования:

$$B \equiv B \vee 0 \equiv B \vee x_i \bar{x}_i \equiv (B \vee x_i)(B \vee \bar{x}_i)$$

2. Если в КНФ A входят две одинаковые элементарные дизъюнкции, то используя следующее равносильное преобразование:

$$BB \equiv B,$$

одну элементарную дизъюнкцию можно отбросить.

3. Если элементарная дизъюнкция B КНФ A содержит одновременно переменную x_i и ее отрицание, то используя следующие равносильные преобразования:

$$B \vee x_i \vee \bar{x}_i \equiv B \vee 1 = 1,$$

эту элементарную дизъюнкцию можно отбросить.

4. Если элементарная дизъюнкция B КНФ A содержит дважды переменную x_i , то используя следующее равносильное преобразование:

$$B \vee x_i \vee x_i \equiv B \vee x_i,$$

одну переменную x_i можно отбросить.

СКНФ формулы A существует в единственном виде.

ПРИМЕР

Получить СКНФ формулы $A = x \rightarrow y\bar{z}$

С помощью равносильных преобразований получаем СКНФ A :

$$\begin{aligned} A = x \rightarrow y\bar{z} &\equiv \bar{x} \vee y\bar{z} \equiv (\bar{x} \vee y)(\bar{x} \vee \bar{z}) \equiv (\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{z} \vee y)(\bar{x} \vee \bar{z} \vee \bar{y}) \equiv \\ &\equiv (\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee \bar{z}) \end{aligned}$$

С помощью таблицы истинности получаем СДНФ A :

x	y	z	\bar{z}	$y\bar{z}$	$A = x \rightarrow y\bar{z}$	\bar{A}
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	1	0	1	1	1	0
0	1	1	0	0	1	0
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	1	0	1	1	1	0
1	1	1	0	0	0	1

$$СДНФ \bar{A} = x\bar{y}\bar{z} \vee x\bar{y}z \vee xyz$$

$$СКНФ A \equiv \overline{СДНФ \bar{A}} = \overline{x\bar{y}\bar{z} \vee x\bar{y}z \vee xyz} \equiv (\bar{x} \vee y \vee z)(\bar{x} \vee y \vee \bar{z})(\bar{x} \vee \bar{y} \vee \bar{z})$$

Очевидно, что в результате двух способов совпадает.

СДНФ формулы A можно получить из СКНФ A , используя следующее правило:

$$\text{СДНФ } A = \overline{\text{СКНФ } \bar{A}}$$

Выполнимость формулы алгебры логики

Все формулы алгебры логики делятся на три класс:

1. **тождественно истинные** или **тавтологии**;
2. **тождественно ложные** или **противоречия**;
3. **выполнимые**.

Формулу называют **тождественно истинной (тавтологией)**, если она принимает значение истинно при любых значениях входящих в нее переменных.

Формула называется **тождественно ложной (противоречие)**, если она принимает значение ложно при любых значениях переменных входящих в нее.

Формула A называется **выполнимой**, если она принимает значение истинно хотя бы один раз на любом наборе значений входящих в нее переменных и не является тождественно истинной..

Тождественно истинная формула не имеет СКНФ, а ложная СДНФ.

ПРИМЕР

Формула $A = x\bar{x}$ является тождественно ложной:

x	\bar{x}	$A = x\bar{x}$
0	1	0
1	0	0

Формула $A = x \vee \bar{x}$ является тождественно истинной:

x	\bar{x}	$A = x \vee \bar{x}$
0	1	1
1	0	1

Формула $A = x \vee y$ является выполнимой:

x	y	$A = x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

Очевидно, что тождественно ложная формула не имеет СДНФ, а тождественно истинная – СКНФ. Выполнимая формула алгебры логики имеет СДНФ и СКНФ,

Применение математической логики.

С помощью алгебры логики можно:

- решать логические задачи;
- реализация технических устройств.

С помощью алгебры логики можно решать логические задачи. Суть применения методов алгебры логики к решению логических задач состоит в том, что, имея конкретные условия логической задачи, необходимо записать их в виде формулы логики. В дальнейшем путем равносильных преобразований упрощают полученную формулу. Простейший вид формулы, как правило, приводит к ответу на все вопросы задачи.

ПРИМЕР

Определить, Был ли Смит убийцей, если известно следующее:

Если Джонс не встречал этой ночью Смита, то либо Смит был убийцей, либо Джонс лжет. Если Смит не был убийцей, то Джонс не встречал Смита этой ночью, и убийство имело место после полуночи. Если убийство было совершено после полуночи, то либо Смит был убийцей, либо Джонс лжет.

Составим элементарные высказывания:

A – Джонс не встречал Смит этой ночью.

B – Смит был убийца.

C – Джонс лжет.

D – убийство было совершено после полуночи.

Тогда сложные высказывания можно записать на языке алгебры логики в следующем виде:

Если Джонс не встречал этой ночью Смита, то либо Смит был убийцей, либо Джонс лжет. - $A \rightarrow (B \vee C)$

Если Смит не был убийцей, то Джонс не встречал Смита этой ночью, и убийство имело место после полуночи. - $\bar{B} \rightarrow AD$

Если убийство было совершено после полуночи, то либо Смит был убийцей, либо Джонс лжет. - $D \rightarrow (B \vee C)$

Вся картина преступления может быть представлена в виде формулы:

$$(A \rightarrow (B \vee C)) \cdot (\bar{B} \rightarrow AD) \cdot (D \rightarrow (B \vee C))$$

Упростим полученную формулу с помощью равносильных преобразований:

$$\begin{aligned} (A \rightarrow (B \vee C)) \cdot (\bar{B} \rightarrow AD) \cdot (D \rightarrow (B \vee C)) &\equiv (\bar{A} \vee B \vee C)(\bar{\bar{B}} \vee AD)(\bar{D} \vee B \vee C) \equiv \\ &\equiv (\bar{A} \vee B \vee C)(B \vee AD)(B \vee C \vee \bar{D}) \equiv (\bar{A}B \vee B \vee ABD \vee BC \vee ACD)(B \vee C \vee \bar{D}) \equiv \\ &\equiv (B \vee ACD)(B \vee C \vee \bar{D}) \equiv B \vee BC \vee B\bar{D} \vee ABCD \vee ACD \equiv B \vee ACD \end{aligned}$$

Ответ $B \vee ACD$ трактуется так: Либо Смит был убийцей, либо убийство совершено после полуночи Джонс лжет, что видел Смита этой ночью. Последнее сложное высказывание (ACD) по сути также определяет вину Смита, как и прямое утверждение, что Смит был убийцей (B). Таким образом приговор Смицу вынесен.

В техническом аспекте математическая логика применяется в технических средствах автоматизации в виде релейно-контактных схем (РКС) и логических элементов «и-не», «или-не».

РКС представляют собой переключатели, которые могут находиться либо замкнутом состоянии (1), либо в разомкнутом состоянии (0).

Логические элементы «и-не» осуществляют логическую функцию:

$$y = \overline{x_1 x_2 \dots x_n}$$

Логические элементы «или-не» осуществляют логическую функцию:

$$y = \overline{x_1 \vee x_2 \vee \dots \vee x_n}$$

Нетрудно заметить, что суть и РКС, и логических элементов – это формулы математической логики. Суть применения методов алгебры логики к конструированию технических средств автоматизации на базе РКС технического устройства, необходимо записать принцип его работы в виде формулы логики. В дальнейшем путем равносильных преобразований упрощают полученную формулу. Далее полученную формулу реализуют на базе РКС или логических элементов.

Важным требованием при конструировании технических средств автоматизации является минимальное количество базовых элементов (РКС, логических элементов). Поэтому задача минимизации сложных высказываний является особо актуальной.

Минимизация сложных высказываний.

Существует несколько способов минимизации сложных высказываний. Рассмотрим самые распространенные:

- метод Квайна;
- карты Вейча;
- минимизирующие карты.

Метод Квайна.

Алгоритм метода Квайна включает в себя следующие этапы:

1. Любая формула A приводится к СДНФ.
2. СДНФ приводится к сокращенной ДНФ (СкДНФ). При получении СкДНФ используются следующие формулы равносильности:

а) Формула склеивания

$$AB \vee A\bar{B} \equiv A$$

б) Формула неполного склеивания

$$AB \vee A\bar{B} \equiv A \vee AB \vee A\bar{B}$$

в) Формула поглощения

$$A \vee AB \equiv A$$

Применяя все возможные процедуры склеивания, СДНФ приводится к СкДНФ.

3. Минимальная форма формулы A (МДНФ A) получается на основе **импликантной матрицы** путем нахождения минимального покрытия этой матрицы. **Импликанта** – это элементарная конъюнкция СкДНФ. **Конституента единицы** – это элементарная конъюнкция СДНФ. Импликантная матрица – это матрица **импликант** и **конституент единицы**. (столбцы - конституенты единицы, строки – импликанты). МДНФ может быть несколько.

ПРИМЕР.

Необходимо найти МДНФ формулы:

$$\begin{array}{cccccc} \overline{X}\overline{Y}\overline{Z} & \vee & \overline{X}\overline{Y}Z & \vee & \overline{X}YZ & \vee & X\overline{Y}\overline{Z} & \vee & X\overline{Y}Z & \vee & XYZ \\ 1 & & 2 & & 3 & & 4 & & 5 & & 6 \end{array}$$

Осуществляем всевозможные склеивания

$$1-2 \quad \overline{X}\overline{Y}\overline{Z} \vee \overline{X}\overline{Y}Z \equiv \overline{X}\overline{Y}$$

$$1-4 \quad \overline{Y}\overline{Z}$$

$$2-3 \quad \overline{X}Z$$

3-6 YZ

4-5 $X\bar{Z}$

5-6 XY

СкДНФ имеет вид:

$$\overline{XY} \vee \overline{YZ} \vee \overline{XZ} \vee YZ \vee X\bar{Z} \vee XY$$

Составляем импликантную матрицу

	\overline{XYZ}	$\overline{X\bar{Y}Z}$	$\overline{XY\bar{Z}}$	$X\bar{Y}\bar{Z}$	$XY\bar{Z}$	XYZ
\overline{XY}	+	+				
\overline{YZ}	+			+		
\overline{XZ}		+	+			
YZ			+			+
$X\bar{Z}$				+	+	
XY					+	+

По данной импликантной матрице можно выбрать следующие МДНФ

$$\overline{XY} \vee YZ \vee X\bar{Z}$$

$$\overline{YZ} \vee \overline{XZ} \vee XY$$

Метод минимизирующих карт.

Алгоритм метода минимизирующих карт включает в себя следующие этапы:

1. Любая формула приводится к СДНФ.
2. Составляется таблица всевозможных сочетаний переменных.
3. Из таблицы вычеркиваются те строки, которые не содержат конstituенты СДНФ. Конъюнкции этих строк вычеркиваются в других строках.
4. В каждой строке оставляются конъюнкции с минимальным количеством переменных.
5. Из каждой строки выбирается одна конъюнкция и составляется ДНФ.
6. Из построенных ДНФ выбирается минимальная.

ПРИМЕР

Дана СДНФ

$$\overline{XYZ} \vee \overline{X\bar{Y}Z} \vee \overline{XY\bar{Z}} \vee X\bar{Y}\bar{Z} \vee XY\bar{Z} \vee XYZ$$

X	Y	Z	XY	$X\bar{Z}$	YZ	XYZ	
X	Y	\bar{Z}	XY	$X\bar{Z}$	YZ	XYZ	
X	\bar{Y}	Z	$X\bar{Y}$	XZ	$\bar{Y}Z$	$X\bar{Y}Z$	*
X	\bar{Y}	\bar{Z}	$X\bar{Y}$	$X\bar{Z}$	$\bar{Y}\bar{Z}$	$X\bar{Y}\bar{Z}$	
\bar{X}	Y	Z	$\bar{X}Y$	$\bar{X}Z$	YZ	$\bar{X}YZ$	
\bar{X}	Y	\bar{Z}	$\bar{X}Y$	$\bar{X}\bar{Z}$	$Y\bar{Z}$	$\bar{X}Y\bar{Z}$	*
\bar{X}	\bar{Y}	Z	$\bar{X}\bar{Y}$	$\bar{X}Z$	$\bar{Y}Z$	$\bar{X}\bar{Y}Z$	
\bar{X}	\bar{Y}	\bar{Z}	$\bar{X}\bar{Y}$	$\bar{X}\bar{Z}$	$\bar{Y}\bar{Z}$	$\bar{X}\bar{Y}\bar{Z}$	

* - помечены строки, не содержащие конstituенты СДНФ.

После соответствующих преобразований получаем следующую таблицу

			\overline{XY}		\overline{YZ}		
			XY	$X\overline{Z}$			
							*
				\overline{XZ}	\overline{YZ}		
				\overline{XZ}	YZ		
							*
			$\overline{X\overline{Y}}$	\overline{XZ}			
			$\overline{X\overline{Y}}$		$\overline{Y\overline{Z}}$		

После всевозможного перебора остаются следующие МДНФ:

$$\overline{X\overline{Y}} \vee YZ \vee X\overline{Z}$$

$$\overline{Y\overline{Z}} \vee \overline{XZ} \vee XY$$

Метод минимизации с помощью карт Вейча.

Алгоритм метода карт Вейча включает в себя следующие этапы:

1. Любая формула приводится к СДНФ.
2. Составляется карта Вейча. Карта Вейча – это таблица всех возможных комбинаций значений переменных. В соответствующие ячейки заносятся единицы, соответствующие конъюнктам СДНФ.
3. Единицы, стоящие по вертикали и горизонтали, объединяются (по 2, по 4, по 8 и т.д.). Объединение единиц соответствует операциям склеивания и поглощения. Иначе говоря, формируются максимальные подкубы.
4. Для каждого объединения выписываются конъюнкции из элементов, общих для каждой единицы, входящих в объединение..
5. Выше полученные конъюнкции составляют МДНФ.

Карты Вейча удобны при поиске МДНФ функций двух, трех и четырех переменных.

n=2

$$\text{СДНФ} = XY \vee X\overline{Y} \vee \overline{X}\overline{Y}$$

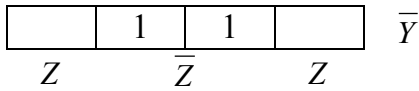
	X	\overline{X}	
Y	1	0	1
\overline{Y}	1	0	1

$$\text{МДНФ} = X \vee \overline{Y}$$

n=3

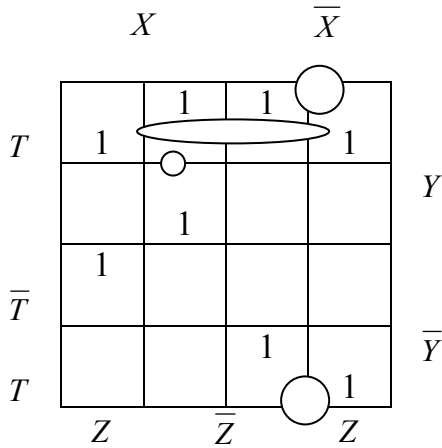
$$\text{СДНФ} = XYZ \vee XY\overline{Z} \vee X\overline{Y}\overline{Z} \vee \overline{X}\overline{Y}\overline{Z}$$

	X	\overline{X}	
Y	1	0	1
\overline{Y}	1	0	1



МДНФ = $XY \vee \bar{Y}\bar{Z}$
 n=4

СДНФ = $XYZT \vee XYZ\bar{T} \vee \bar{X}YZT \vee \bar{X}YZ\bar{T} \vee X\bar{Y}ZT \vee X\bar{Y}Z\bar{T} \vee \bar{X}\bar{Y}ZT \vee \bar{X}\bar{Y}Z\bar{T}$



МДНФ = $YT \vee \bar{X}T \vee XY\bar{Z} \vee X\bar{Y}Z\bar{T}$

Булевы функции и их свойства.

Булевой функцией называется функция n переменных, которая принимает значение 1 или 0, а так же ее аргументы тоже принимают значение 1 или 0.

Булевая функция имеет следующие свойства:

1. **Свойство сохранения нуля T_0 .** Булевая функция сохраняет ноль, если функция при нулевых значениях аргумента принимает значение ноль.
2. **Свойство сохранения единицы T_1 .** Булевая функция сохраняет единицу, если функция при единичных значениях аргумента принимает значение единица.

ПРИМЕР

Логическая операция – дизъюнкция $F = x \vee y$ обладает и свойством сохранения нуля ($F(0,0) = x \vee y = 0$), и свойством сохранения единицы ($F(1,1) = x \vee y = 1$)

3. **Линейность L.** Функция является линейной, если её можно представить в виде:

$$F(x_1, x_2, \dots, x_n) \equiv a_0 \oplus a_1 x_1 \oplus \dots \oplus a_n x_n$$

где a_i - булевая переменная

ПРИМЕР

Эквивалентность является линейной функцией:

$$x \leftrightarrow y \equiv 1 \oplus 1 \cdot x \oplus 1 \cdot y \equiv 1 \oplus x \oplus y$$

4. **Монотонность M.** Функция является монотонной, если для любых произвольных наборов α и β выполняются следующие неравенства:

$$\alpha < \beta$$

$$F(\alpha) < F(\beta)$$

5. **Самодвойственность S.** Функция называется самодвойственной, если она равна двойственной ей функции.

Двойственной функцией называется функция:

$$F^*(x_1, x_2, \dots, x_n) = \overline{\overline{F(\overline{x_1}, \overline{x_2}, \dots, \overline{x_n})}}$$

Тогда свойство самодвойственности может представлено:

$$F(x_1, x_2, \dots, x_n) = F^*(x_1, x_2, \dots, x_n)$$

ПРИМЕР

Отрицание является самодвойственной функцией:

$$F(x) = F^*(x) = \overline{\overline{x}} = \overline{\overline{\overline{x}}} = \overline{x}$$

Функциональная полнота. Теорема Поста.

Функциональный набор логических функций – это такой набор функций, который позволяет любую функцию математической логики описать с помощью функций данного набора.

Теорема Поста. Для того чтобы набор функций $\{f_1, f_2, \dots, f_n\}$ был функционально полный необходимо и достаточно, чтобы для всего набора функций в целом не выполнялись свойства сохранения нуля, сохранения единицы, линейности, монотонности и самодвойственности.

Полноту набора удобно определять по таблице Поста, в клетках которой ставится знак «+» или «-» в зависимости от того, обладает функция из этого набора тем или иным свойством. В силу теоремы Поста для полноты системы необходимо и достаточно, чтобы в каждом столбце был хотя бы один минус.

{ }	T ₀	T ₁	L	M	S
F ₁	+	+	+	+	-
F ₂	-	-	-	-	+

ПРИМЕР

Доказать, что набор функций дизъюнкция и отрицание является функционально полным.

	T ₀	T ₁	L	M	S
\neg	-	-	+	-	+
V	+	+	-	+	-

Логика предикат.

Предикат – это сложное высказывание, в котором аргументы (x_1, x_2, \dots, x_n) принимают значение из некоторой вещественной области D , а значение самого высказывания принимает значение истинно или ложно.

$$P(x_1, x_2, \dots, x_n)$$

$$x_i \in D$$

$$P(x_1, x_2, \dots, x_n) \in \{0,1\}$$

ПРИМЕР

Предикатом является высказывание – быть четным числом на множестве натуральных чисел:

$P(x)$ - быть четным числом;

$$x \in N = \{1,2,3,\dots\}$$

$$P(x) \in \{0,1\}$$

$$P(5) = 0$$

$$P(10) = 1$$

Предикаты могут быть одноместные - $P(x)$, двухместные - $P(x, y)$ ($x \in D_1, y \in D_2$) и
 многоместные - $P(x_1, x_2, \dots, x_n)$ ($x_1 \in D_1, x_2 \in D_2, \dots, x_n \in D_n$)

Логические операции над предикатами.

Для предикатов выполнимы следующие операции:

Конъюнкция $P(x) \cdot Q(x)$ - это новый предикат, который принимает значение истинно при тех и только тех значениях x из вещественной области D , при которых оба предиката $P(x)$ и $Q(x)$ истинны одновременно, и ложно во всех других случаях.

Дизъюнкция $P(x) \vee Q(x)$ - это новый предикат, который принимает значение ложно при тех и только тех значениях x из вещественной области D , при которых оба предиката $P(x)$ и $Q(x)$ ложны одновременно, и истинно во всех других случаях.

Отрицание предиката $\overline{P(x)}$ - это новый предикат, который принимает значение истинно при всех x из вещественной области D , при которых предикат $P(x)$ принимает значение ложно и наоборот.

Квантовые операции.

Для предикатов кроме логических операций применимы **кванторные операции: всеобщности и существования.**

Пусть $P(x)$ - предикат, определенный на множестве D . Тогда $\forall x P(x)$ - означает «для всякого (любого) x истинно $P(x)$ ». Символ \forall называется **квантором всеобщности.**

Переменную x в предикате $P(x)$ называют **свободной** (ей можно придавать различные значения из M), в высказывании $\forall x P(x)$ переменную x называют **связанной** квантором всеобщности.

Пусть $P(x)$ - предикат, определенный на множестве D . Тогда $\exists xP(x)$ - означает «существует x , для которого истинно $P(x)$ ». Символ \exists называется **квантором существования**.

ПРИМЕР

Пусть на множестве натуральных чисел задан предикат $P(x)$ - «число x кратно 3». Используя кванторы, из данного предиката можно получить высказывания:

$\forall xP(x)$ - «все натуральные числа кратны 3»;

$\exists xP(x)$ - «существуют натуральные числа, кратные 3».

Очевидно, что первое из данных высказываний ложно, а второе – истинно.

Кванторные операции применяются и к **многоместным предикатам**. Пусть на множестве D задан **двухместный предикат** $P(x, y)$. К данному предикату могут применяться кванторные операции как по одной, так и по двум переменным.

ПРИМЕР

Пусть предикат $P(x, y)$ означает x делится на y без остатка., причем обе переменные определены на множестве натуральных чисел. Тогда применение кванторных операций приводит к следующим высказываниям:

1. $\forall x \forall y P(x, y)$ - «для любого x и для любого y справедливо, что x делится на y без остатка.

2. $\forall x \exists y P(x, y)$ - «для любого x существует y , который является делителем x без остатка.

Нетрудно заметить, что первое высказывание является ложным, а второе – истинным.

Для **многоместных предикатов**, связанных по одной переменной справедливы следующие формулы:

$$\forall x_1 P(x_1, x_2, \dots, x_n) = P(\alpha_1, x_2, \dots, x_n) \cdot P(\alpha_2, x_2, \dots, x_n) \cdot \dots \cdot P(\alpha_m, x_2, \dots, x_n),$$

где $x_1 \in \{\alpha_1, \alpha_2, \dots, \alpha_m\}$

$$\exists x_1 P(x_1, x_2, \dots, x_n) = P(\alpha_1, x_2, \dots, x_n) \vee P(\alpha_2, x_2, \dots, x_n) \vee \dots \vee P(\alpha_m, x_2, \dots, x_n),$$

где $x_1 \in \{\alpha_1, \alpha_2, \dots, \alpha_m\}$

Равносильные формулы логики предикатов.

Две формулы логики предикатов A и B называются **равносильными на области D** , если они принимают одинаковые логические значения при всех значениях входящих в них переменных, отнесенных к области D .

Две формулы логики предикатов A и B называются **равносильными**, если они равносильны во всякой области.

Очевидно, что все формулы равносильности алгебры высказываний будут верны, если в них вместо переменных подставить формулы логики предикатов. Но, кроме того, имеют место равносильности самой логики предикатов:

Закон де Моргана:

$$1. \overline{\forall x A(x)} \equiv \exists x \overline{A(x)}$$

$$2. \overline{\exists x A(x)} \equiv \forall x \overline{A(x)}$$

Закон двойного отрицания:

$$3. \forall x A(x) \equiv \overline{\exists x \overline{A(x)}}$$

$$4. \exists x A(x) \equiv \overline{\forall x \overline{A(x)}}$$

$$5. \forall x A(x) \cdot \forall x B(x) \equiv \forall x [A(x) \cdot B(x)]$$

$$6. \exists x A(x) \vee \exists x B(x) \equiv \exists x [A(x) \vee B(x)]$$

$$7. \exists x A(x) \cdot \exists y B(y) \equiv \exists x \exists y [A(x) \cdot B(y)]$$

Для произвольного высказывания C (предиката, не связанного по переменной x) справедливы следующие формулы равносильности:

$$8. C \cdot \forall x A(x) \equiv \forall x [C \cdot A(x)]$$

$$9. C \vee \forall x A(x) \equiv \forall x [C \vee A(x)]$$

$$10. C \cdot \exists x A(x) \equiv \exists x [C \cdot A(x)]$$

$$11. C \vee \exists x A(x) \equiv \exists x [C \vee A(x)]$$

$$12. \forall x C \equiv C$$

$$13. \exists x C \equiv C$$

Формулы замены переменных (где x и y из одной предметной области):

$$14. \forall x A(x) \equiv \forall y A(y)$$

$$15. \exists x A(x) \equiv \exists y A(y)$$

Предваренная нормальная форма предиката

Формула предиката имеет **нормальную форму**, если она содержит только операции конъюнкции, дизъюнкции, и кванторные операции, а операция отрицания отнесена к элементарному предикату.

ПРИМЕР

Пусть дан предикат $(\exists x P(x) \rightarrow \forall y Q(y)) \rightarrow R(z)$. Привести данный предикат к нормальной форме

$$\begin{aligned} (\exists x P(x) \rightarrow \forall y Q(y)) \rightarrow R(z) &\equiv \overline{(\exists x P(x) \rightarrow \forall y Q(y))} \vee R(z) \equiv \\ &\equiv \overline{(\exists x P(x) \vee \forall y \overline{Q(y)})} \vee R(z) \equiv \overline{\exists x P(x)} \cdot \overline{\forall y \overline{Q(y)}} \vee R(z) \equiv \\ &\equiv \exists x \overline{P(x)} \cdot \exists y \overline{\overline{Q(y)}} \vee R(z) \end{aligned}$$

Предварённой нормальной формой предиката называется такая нормальная форма предиката, в которой кванторные операторы либо отсутствуют, либо используются после операций алгебры логики.

ПРИМЕР

Привести предикат из предыдущего примера к предваренной нормальной форме предиката

$$\begin{aligned} \exists x \overline{P(x)} \cdot \exists y \overline{\overline{Q(y)}} \vee R(z) &\equiv \exists x \exists y (\overline{P(x)} \cdot \overline{\overline{Q(y)}}) \vee R(z) \equiv \\ &\equiv \exists x \exists y (\overline{P(x)} \cdot \overline{Q(y)}) \vee R(z) \end{aligned}$$

ТЕОРИЯ ГРАФОВ

Начало теории графов как математической дисциплины было положено Леонардом Эйлером в его знаменитом решении задачи о Кенигсбергских мостах в 1736 году. План города Кенигсберга представлен на рис. 3.1. Задача о Кенигсбергских мостах сводилась к тому, чтобы построить маршрут своей воскресной прогулки так, чтобы, начиная в любой точке суши (А, В, С или D) пройти по всем мостам строго по одному разу и вернуться в исходную точку (начало маршрута).

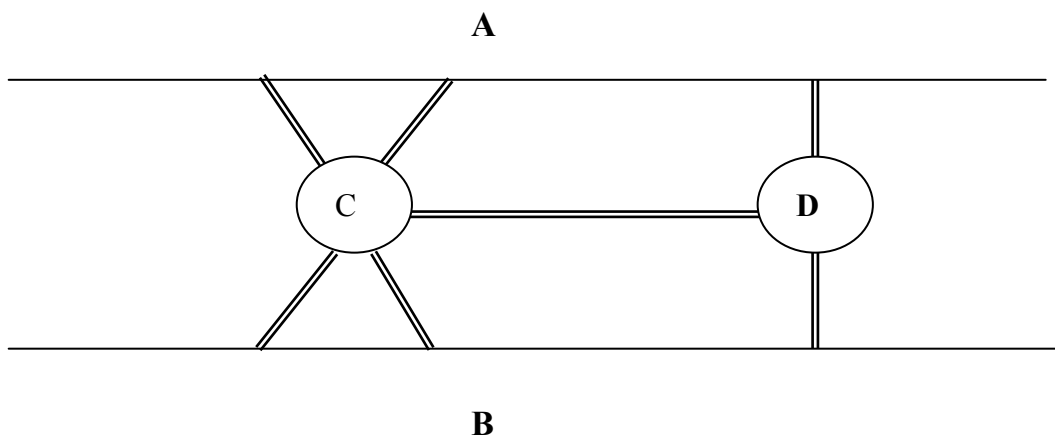


Рис. 3.1. Иллюстрация к задаче о Кенигсбергских мостах.

Такую задачу не предоставляется возможным решить классическими методами математики. Для решения такой задачи был предложен качественно новый аппарат – аппарат теории графов.

Основные понятия теории графов

Графом называется пара следующего вида:

$$G = \langle \Gamma, X \rangle, \tag{3.1}$$

где Γ - график $\Gamma \subseteq X \times X$;

X - множество вершин.

Иными словами, **граф** представляет совокупность множества вершин и дуг.

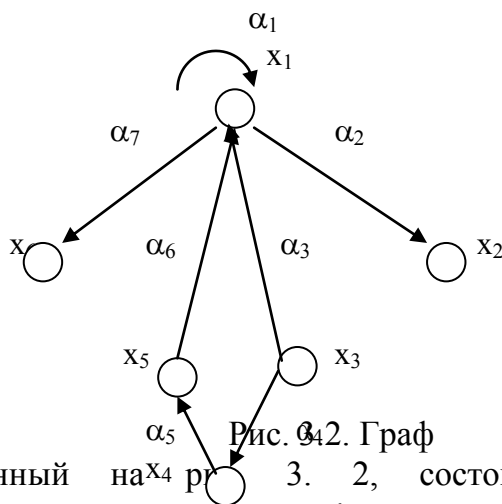


Рис. 3.2. Граф

Граф, представленный на рис. 3.2, состоит из множества вершин $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ и множество дуг $\alpha = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8\}$

Графическое изображение графа является самым наглядным, но не единственным способом задания графа. Кроме того граф может быть задан:

1. перечислением:

$$\{ \langle x_1, x_1 \rangle, \langle x_1, x_2 \rangle, \langle x_1, x_6 \rangle, \langle x_3, x_1 \rangle, \langle x_3, x_4 \rangle, \langle x_4, x_5 \rangle, \langle x_5, x_1 \rangle \}$$

2. множеством образов:

$$\Gamma_{x_1} = \{x_1, x_2, x_6\}$$

$$\Gamma_{x_2} = \emptyset$$

$$\Gamma_{x_3} = \{x_1, x_4\}$$

$$\Gamma_{x_4} = \{x_5\}$$

$$\Gamma_{x_5} = \{x_1\}$$

$$\Gamma_{x_6} = \emptyset$$

где Γ_{x_i} - образ вершины x_i - множество вершин, в которые исходят дуги из данной вершины.

3. матрицей инцидентности

Матрица инцидентности - это матрица вершин и инцидентных им дуг.

Дуга **инцидентна** вершине, если эта дуга исходит или заходит в данную вершину.

Вершина **инцидентна** дуге, если в эту вершину заходит или исходит данная дуга.

В матрице инцидентности в первом столбце расположены вершины, в первой строке – дуги. Остальные ячейки матрицы инцидентности заполняются по следующему правилу:

- $\gamma_{ij} = -1$, если из i -той вершины исходит j -тая дуга;
- $\gamma_{ij} = +1$, если в i -той вершину заходит j -тая дуга;
- $\gamma_{ij} = 0$, если i -тая вершина не инцидента j -той дуге;
- $\gamma_{ij} = 2$, если из i -той вершины исходит j -тая дуга и в нее же заходит, т.е. в i -той вершине j -тая дуга образует **петлю**.

Для графа, представленного на рис. 3.2 матрица инцидентности имеет вид:

	α_1	α_2	α_3	α_4	α_5	α_6	α_7
x_1	2	-1	+1	0	0	+1	-1
x_2	0	+1	0	0	0	0	0
x_3	0	0	-1	-1	0	0	0
x_4	0	0	0	+1	-1	0	0

x_5	0	0	0	0	+1	-1	0
x_6	0	0	0	0	0	0	+1

На практике в матрице инцидентности иногда нули не проставляются для наглядности.

	α_1	α_2	α_3	α_4	α_5	α_6	α_7
x_1	2	-1	+1			+1	-1
x_2		+1					
x_3			-1	-1			
x_4				+1	-1		
x_5					+1	-1	
x_6							+1

Свойство матрицы инцидентности – сумма элементов по столбцам равна 0 или 2.

4. матрицей смежности

Смежные дуги – это дуги инцидентные одной вершине.

Смежные вершины – вершины, инцидентные одной дуге.

Матрица смежности - это матрица смежных вершин.

Матрица смежности заполняется по следующему правилу: $\gamma_{ij} = 1$, если из i -той вершины исходит дуга в j – тую вершину; во всех остальных случаях $\gamma_{ij} = 0$ (для удобства и наглядности на практике в матрице смежности нули не проставляются).

Для графа, представленного на рис. 3.2 матрица смежности имеет вид:

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1	1				1
x_2						
x_3	1			1		
x_4					1	
x_5	1					
x_6						

Полустепенью захода $\rho^+(x_i)$ i -той вершины называется число дуг, заходящих в данную вершину.

Полустепенью исхода $\rho^-(x_i)$ i -той вершины называется число дуг, исходящих из данной вершины.

Степенью i -той вершины исхода $\rho(x_i)$ называется сумма полустепеней захода и исхода:

$$\rho(x_i) = \rho^-(x_i) + \rho^+(x_i) \quad (3.2)$$

Свойства матрицы смежности:

1. Сумма единиц по строке определяет полустепень исхода;
2. Сумма единиц по столбцу определяет полустепень захода;
3. Сумма единиц по строкам и по столбцам определяет степень вершин.

Основное свойство графа:

В любых графах число вершин с нечетной степенью четно.

Путем в графе называется последовательность дуг такая, что каждая следующая дуга исходит из вершины, в которую заходит предыдущая дуга.

Длиной пути называется количество пройденных дуг.

Простой путь – это такой путь, в котором дуга встречается не более одного раза.

Элементарный путь – это путь, в котором вершина встречается не более одного раза.

Контур – путь, начинающийся и заканчивающийся в одной точке.

Петля – контур длиной в одну единицу.

Графы бывают двух видов:

- **ориентированный граф (орграф)** - это граф, состоящий из вершин и дуг.
- **неориентированный граф** – это граф, состоящий из вершин и ребер. **Ребро** – это дуга, не имеющая направление.

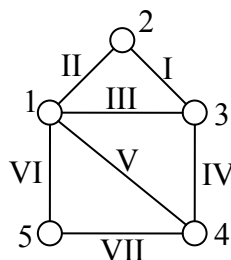


Рис. 3.3. Неориентированный граф

В неориентированном графе путь называется **цепью**; контур – **циклом**.

Неориентированный граф также может быть задан с помощью матриц инцидентности и смежности.

Матрица инцидентности для неориентированного графа составляется по правилу:

- $\gamma_{ij} = 1$, если i -тая вершина инцидентна j -тому ребру;
- $\gamma_{ij} = 0$, если i -тая вершина не инцидентна j -тому ребру;
- $\gamma_{ij} = 2$, если в i -той вершине j -тое ребро образует петлю.

Для графа, представленного на рис. 3.3, матрица инцидентности имеет вид:

	I	II	III	IV	V	VI	VII
1		1	1		1	1	
2	1	1					
3	1		1	1			
4				1	1		1
5						1	1

Матрица смежности для неориентированного графа составляется по правилу: $\gamma_{ij} = 1$, если из i -тая и j – тая вершины смежные.

Для графа, представленного на рис. 3.3, матрица смежности имеет вид:

	1	2	3	4	5
1	1	1	1	1	1
2	1	1			
3	1	1		1	
4	1		1		1
5	1			1	

Матрица смежности для неориентированного графа симметрична относительно главной диагонали. Степень i - той вершины равна сумме элементов по строке или по столбцу матрицы смежности.

Если в графе присутствуют как ребра, так и дуги, то он называется **смешанным**.

Если между двумя вершинами существует более чем 1 дуга или ребро, то такой граф называется **мультиграфом**.

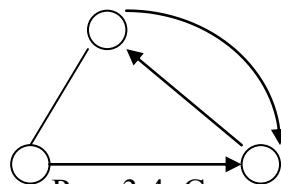


Рис. 3.4. Смешанный мультиграф.

Граф называется **связным**, если между любыми двумя вершинами которого существует путь (цепь).

Эйлеров граф.

Эйлеровой цепью называется цепь, проходящая по всем ребрам графа.

Эйлеровым циклом называется эйлерова цепь, начинающаяся и заканчивающаяся в одной вершине.

Эйлеровым графом называется граф, содержащий Эйлеров цикл.

ТЕОРЕМА: Для того чтобы связанный граф был Эйлеровым, необходимо и достаточно, чтобы степени всех вершин были четны

Данная теорема позволяет решить задачу о Кенигсбергских мостах. Граф, соответствующий данной задаче, имеет вид

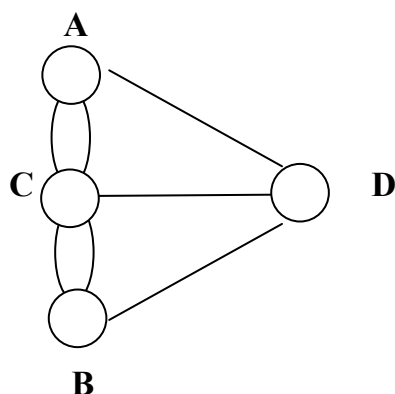


Рис. 3.5 Граф «Кенигсбергские мосты»

Вершины графа (A, B, C, D) имеют степени:

$$\begin{aligned} \rho_A &= 3 \\ \rho_B &= 3 \\ \rho_C &= 5 \\ \rho_D &= 3 \end{aligned} \tag{3.3}$$

Тогда в соответствии с теоремой Эйлера данный граф не является Эйлеровым. А это означает, что нельзя обойти все мосты г. Кенигсберга строго по одному разу, начав и закончив путь в одной точке суши.

Следствие из ТЕОРЕМЫ: Для того чтобы граф содержал Эйлеровую цепь, необходимо и достаточно, чтобы две вершины имели нечетную степень. При этом в одной из вершин цепь будет начинаться, в другой – заканчиваться.

Множество внутренней устойчивости графа

Множество внутренней устойчивости графа – это множество несмежных вершин.

Пусть дан граф $G = \langle \Gamma, X \rangle$. Тогда для множества внутренней устойчивости X' справедливо следующее:

$$\begin{aligned} X' &\subseteq X \\ X' \cap \Gamma_{X'} &= \emptyset \end{aligned} \tag{3.4}$$

Максимальным множеством внутренней устойчивости называется внутренне устойчивое множество, добавление любой вершины к которому, делает это множество неустойчивым.

ПРИМЕР

Дан граф

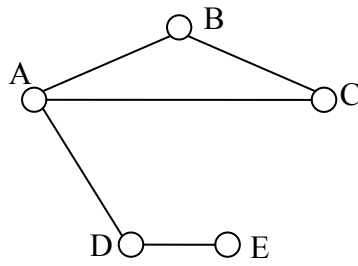


Рис. 3.6 Граф

Данный граф имеет следующие множества внутренней устойчивости. $\{A, E\}, \{B, D\}, \{B, E\}, \{C, D\}, \{C, E\}$

Рассмотрим множество $\{A, E\}$. Добавляя к нему любую вершину, получаем множества внутренне неустойчивые: $\{A, E, B\}, \{A, E, C\}, \{A, E, D\}$. Следовательно, множество $\{A, E\}$ является максимальным множеством внутренней устойчивости.

Число внутренней устойчивости (α) называется число, равное наибольшей мощности максимального внутренне устойчивого множества.

Очевидно, что поиск максимальных внутренне устойчивых множеств путем простого перебора является неэффективной процедурой, поэтому для поиска максимальных внутренне устойчивых множеств существует специальный алгоритм – **алгоритм Магу**.

Алгоритм Магу для определения множества внутренней устойчивости графа

Пусть дан граф $G = \langle \Gamma, X \rangle$. Для данного графа существует множество внутренней устойчивости X' .

Введем булеву переменную x_i , которая определяется следующим образом:

$x_i = 1$, если вершина x_i принадлежит множеству внутренней устойчивости $x_i \in X'$;

$x_i = 0$, если вершина x_i не принадлежит множеству внутренней устойчивости

$x_i \notin X'$;

Введем булеву переменную α_{ij} :

$\alpha_{ij} = 1$, если между i -той и j -той вершиной есть дуга;

$\alpha_{ij} = 0$, если между i -той и j -той вершиной нет дуги.

Тогда определение внутренней устойчивости (3.4) может быть представлено в следующем виде:

$$\forall i, j (x_i \in X', x_j \in G(x_i)) \Rightarrow x_j \notin X' \quad (3.5)$$

Или используя булеву алгебру:

$$\forall i, j ((x_i \& \alpha_{ij}) \rightarrow \overline{x_j}) = 1 \quad (3.6)$$

Применяя формулы равносильности, преобразуем:

$$\begin{aligned} \forall i, j ((x_i \& \alpha_{ij}) \rightarrow \overline{x_j}) &= \bigwedge_{i=1}^n \bigwedge_{j=1}^n ((x_i \& \alpha_{ij}) \rightarrow \overline{x_j}) = \\ &= \bigwedge_{i=1}^n \bigwedge_{j=1}^n (\overline{(x_i \& \alpha_{ij}) \vee x_j}) = \bigwedge_{i=1}^n \bigwedge_{j=1}^n (\overline{x_i} \vee \overline{\alpha_{ij}} \vee \overline{x_j}) = 1 \end{aligned} \quad (3.7)$$

Рассмотрим выражение:

$$\bigwedge_{i=1}^n \bigwedge_{j=1}^n (\overline{x_i} \vee \overline{\alpha_{ij}} \vee \overline{x_j}) = 1 \quad (3.8)$$

Если $\alpha_{ij} = 0$, то данное равенство является тавтологией.

Если $\alpha_{ij} = 1$, то равенство имеет вид:

$$\alpha_{ij} = 1 \quad \& \quad (\overline{x_i} \vee \overline{x_j}) = 1 \quad (3.9)$$

Данное уравнение лежит в основе алгоритма Магу

Алгоритм Магу состоит из следующих этапов:

1. Для графа составляется матрица смежности.
2. По таблице смежности выписываются все парные дизъюнкции.
3. Выражение приводится к ДНФ.
4. Для любой элементарной конъюнкции выписываются недостающие элементы, которые и образуют множество внутренней устойчивости.

ПРИМЕР

Дан граф

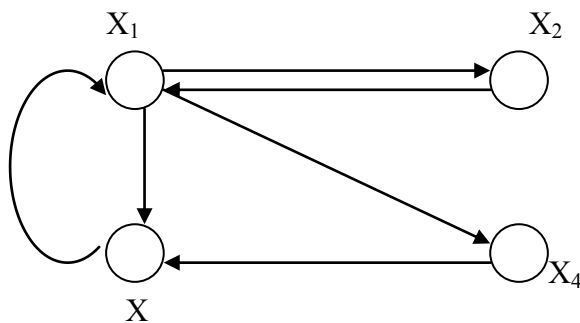


Рис. 3.7 Граф

Матрица смежности имеет вид:

	x_1	x_2	x_3	x_4
x_1		1	1	1
x_2	1			
x_3	1			

x_4			1	
-------	--	--	---	--

Для всех единиц выписываются парные дизъюнкции:

$$(x_1 \vee x_2)(x_1 \vee x_3)(x_1 \vee x_4)(x_2 \vee x_1)(x_3 \vee x_1)(x_4 \vee x_3) \quad (3.10)$$

Приведем выражение к ДНФ:

$$\begin{aligned} & (x_1 \vee x_2)(x_1 \vee x_3)(x_1 \vee x_4)(x_2 \vee x_1)(x_3 \vee x_1)(x_4 \vee x_3) = \\ & = (x_1 \vee x_2)(x_1 \vee x_3)(x_1 \vee x_4)(x_4 \vee x_3) = \\ & = (x_1 \vee x_2 x_3)(x_4 \vee x_1 x_3) = \\ & = x_1 x_4 \vee x_1 x_3 \vee x_2 x_3 x_4 \vee x_1 x_2 x_3 = \\ & = x_1 x_4 \vee x_1 x_3 \vee x_2 x_3 x_4 \end{aligned} \quad (3.11)$$

Множества внутренней устойчивости:

$$\{x_2, x_3\}, \{x_2, x_4\}, \{x_1\}$$

Числом внутренней устойчивости $\alpha = 2$.

Множество внешней устойчивости графа

Множество внешней устойчивости – множество вершин, для которых выполняется одно из следующих правил:

- 1). Любая вершина входит в это множество
- 2) Либо вершина не входит в это множество, но из этой вершины есть дуга в данное множество.

Пусть дан граф $G = \langle \Gamma, X \rangle$. Тогда для множества внешней устойчивости X' справедливо следующее:

$$\begin{aligned} & \forall x_i \in X'' \\ & X'' \cap \Gamma_{X''} \neq \emptyset \end{aligned} \quad (3.12)$$

Число внешней устойчивости (β) – это наименьшая мощность из всех множеств внешней устойчивости.

Алгоритм Магу для определения множества внешней устойчивости.

Пусть дан граф $G = \langle \Gamma, X \rangle$. Для данного графа существует множество внешней устойчивости X'' .

Вводятся булевы переменные x_i и α_{ij} по тому же правилу, что и для алгоритма Магу для определения множества внутренней устойчивости.

Тогда определение множества внешней устойчивости (3.12) запишется следующим образом:

$$\bigwedge_{i=1}^n (x_i \vee \bigvee_{j=1}^n \alpha_{ij} \& x_j) = 1 \quad (3.13)$$

Для $\alpha_{ij} = 1$ справедливо следующее

$$\bigwedge_{i=1}^n (x_i \vee \bigvee_{j=1}^n \alpha_{ij} x_j) = 1 \quad (3.14)$$

Данное уравнение лежит в основе алгоритма Магу

Алгоритм Магу состоит из следующих этапов:

1. Для графа составляется матрица смежности
2. Матрица смежности дополняется единицами (1) по главной диагонали.
3. Для каждой строки выписываются дизъюнкции.
4. Выражение приводится к ДНФ.
5. Все вершины, входящие в элементарную конъюнкцию, образуют множество внешней устойчивости

ПРИМЕР

Определить множество внешней устойчивости для графа, представленного на рис.

3.7.

Матрица смежности имеет вид:

	x_1	x_2	x_3	x_4
x_1		1	1	1
x_2	1			
x_3	1			
x_4			1	

Дополним матрицу смежности единицами по главной диагонали

	x_1	x_2	x_3	x_4
x_1	1	1	1	1
x_2	1	1		
x_3	1		1	
x_4			1	1

Для каждой строки выписываются дизъюнкции:

$$(x_1 \vee x_2 \vee x_3 \vee x_4)(x_1 \vee x_2)(x_1 \vee x_3)(x_3 \vee x_4) \quad (3.15)$$

Приведем выражение к ДНФ:

$$\begin{aligned} & (x_1 \vee x_2 \vee x_3 \vee x_4)(x_1 \vee x_2)(x_1 \vee x_3)(x_3 \vee x_4) = \\ & = (x_1 \vee x_2)(x_1 \vee x_3)(x_3 \vee x_4) = (x_1 \vee x_2 x_3)(x_3 \vee x_4) = \\ & = x_1 x_3 \vee x_1 x_4 \vee x_2 x_3 \vee x_2 x_3 x_4 = x_1 x_3 \vee x_1 x_4 \vee x_2 x_3 \end{aligned} \quad (3.16)$$

Множества внутренней устойчивости:

$$\{x_1, x_3\}, \{x_1, x_4\}, \{x_2, x_3\}$$

Числом внешней устойчивости $\beta = 2$.

Ядром графа называется подмножество вершин, являющихся одновременно внутренней и внешней устойчивостью.

Граф, представленный на рис. 3.7. имеет ядро $\{x_2, x_3\}$

Множество путей в графе

По матрице смежности можно определить, сколько различных путей существует между i -той и j -той вершинами длиной в k единиц. Для этого необходимо определить матрицу M^k , где M - матрица смежности.

Если элемент a_{ij} матрицы M^k :

$a_{ij} = 0$ - между i -той и j -той вершины не существует пути длиной в k единиц;

$a_{ij} = n$ - между i -той и j -той вершины существуют n различных путей длиной в k единиц;

Если M^k - нулевая матрица, это означает, что графе нет путей в k единиц, а максимальный путь – это путь длиной в $(k - 1)$ единиц.

Алгоритм фронта волны. Поиск минимального пути в графе.

Одной из самых распространенных задач в теории графов является задача поиска минимального пути в графе.

Рассмотрим некоторые свойства минимальных путей

1. Любой минимальный путь является простым путем.

2. Если путь $x_1 x_2 \dots x_k$ - минимальный, то любые пути $x_i x_{i+1} \dots x_j$ ($1 \leq i < j \leq k$)

внутри минимального пути также будут минимальны.

Пусть Γ_x^{-1} – прообраз вершины x_i – это множество вершин, из которых исходят дуги в вершину x_i .

Одним из алгоритмов поиска минимального пути в графе является алгоритм фронта волны (FW – *Front Wave*)

Алгоритм фронта волны.

Пусть необходимо найти минимальный путь из вершины x_i в вершину x_j .

1. Выписываются все вершины с 1 по n . Вершина x_i помечается индексом 0.

2. Находится первый фронт волны $FW_1(x_i)$ как множество вершин образа вершины x_i .

$$FW_1(x_i) = \Gamma_{x_i} \quad (3.17)$$

3. Все вершины, принадлежащие первому фронту волны, помечаются индексом 1.

4. Вводится счетчик шагов (фронтов волны) $k=1$.

5. Если $FW_k(x_i) = \emptyset$ или $k=n-1$, то вершина x_j недостижима из вершины, и работа алгоритма на этом заканчивается. В противном смысле переходим к пункту 6.

6. Если $x_j \notin FW_k(x_i)$, то переходим к пункту 8. В противном случае существует путь из вершины x_i в вершину x_j длиной в k единиц, и этот путь минимальный:

$$x_i z_1 z_2 \dots z_{k-1} x_j$$

7. Находятся промежуточные вершины z_1, z_2, \dots, z_{k-1} по правилу:

$$\begin{aligned} z_{k-1} &\in FW_{k-1}(x_i) \cap \Gamma_{x_j}^{-1} \\ z_{k-2} &\in FW_{k-2}(x_i) \cap \Gamma_{z_{k-1}}^{-1}, \\ &\vdots \\ z_1 &\in FW_1(x_i) \cap \Gamma_{z_2}^{-1} \end{aligned} \quad (3.18)$$

где Γ_x^{-1} - **прообраз вершины** x - множество вершин, из которых заходят дуги в вершину x

8. Определяется $k+1$ фронт волны как все непомеченные вершины, принадлежащие образу вершин k -го фронта волны. Помечаются индексом $k+1$ вершины $k+1$ фронта волны. Далее осуществляется переход к пункту 5.

ПРИМЕР

Пусть задан граф матрицей смежности:

	x_1	x_2	x_3	x_4	x_5	x_6
x_1				1	1	
x_2	1			1	1	1
x_3	1	1		1	1	1
x_4		1	1		1	

x_5	1	1	1	1		
x_6	1	1	1	1	1	

Необходимо найти минимальный путь из вершины x_1 в вершину x_6 (по алгоритму «фронта волны»).

1. Выпишем все вершины. Вершина x_1 помечается индексом «0»

$$\begin{array}{cccccc} 0 & & & & & \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array}$$

2. Находится первый фронт волны:

$$FW_1(x_1) = \Gamma_{x_1} = \{x_4, x_5\}$$

3. Все вершины, принадлежащие первому фронту волны, помечаются индексом «1».

$$\begin{array}{cccccc} 0 & & 1 & 1 & & \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array}$$

4. Так как $FW_1(x_1) \neq \emptyset$, $k=1 \neq n-1=6-1=5$ и $x_6 \notin FW_1(x_1)$, то определяем второй фронт волны:

$$\begin{aligned} FW_2(x_1) &= (\Gamma_{x_4} \cup \Gamma_{x_5}) \setminus \{x_1, x_4, x_5\} = \\ &= (\{x_4, x_3, x_5\} \cup \{x_1, x_2, x_3, x_4\}) \setminus \{x_1, x_4, x_5\} = \{x_2, x_3\} \end{aligned}$$

5. Все вершины, принадлежащие второму фронту волны, помечаются индексом «2».

$$\begin{array}{cccccc} 0 & 2 & 2 & 1 & 1 & \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array}$$

6. Так как $FW_2(x_1) \neq \emptyset$, $k=2 \neq n-1=6-1=5$ и $x_6 \notin FW_2(x_1)$, то определяем третий фронт волны:

$$\begin{aligned} FW_3(x_1) &= (\Gamma_{x_2} \cup \Gamma_{x_3}) \setminus \{x_1, x_2, x_3, x_4, x_5\} = \\ &= (\{x_1, x_4, x_5, x_6\} \cup \{x_1, x_2, x_4, x_5, x_6\}) \setminus \{x_1, x_2, x_3, x_4, x_5\} = \\ &= \{x_6\} \end{aligned}$$

7. Так как $x_6 \in FW_3(x_1)$, то существует путь из вершины x_1 в вершину x_6 длиной 3 единицы:

$$x_1 z_1 z_2 x_6$$

8. Находятся промежуточные вершины z_1, z_2 :

$$z_2 \in FW_2(x_1) \cap \Gamma_{x_6}^{-1} = \{x_2, x_3\} \cap \{x_2, x_3\} = \{x_2, x_3\}$$

Выберем $z_2 = x_3$

$$z_1 \in FW_1(x_1) \cap \Gamma_{x_3}^{-1} = \{x_4, x_5\} \cap \{x_4, x_5, x_6\} = \{x_4, x_5\}$$

Выберем $z_1 = x_5$

Таким образом, минимальный путь из вершины x_1 в вершину x_6 имеет вид:

$$x_1 x_3 x_5 x_6$$

Ярусно-параллельная форма графов

Граф, не имеющий контуров, может быть представлен в ярусно-параллельной форме. **Ярусно-параллельная форма** – это такой вид графа, у которого в верхний нулевой ярус помещены вершины, имеющие только исходящие дуги; в нижний ярус помещены вершины, имеющие только входящие дуги. На k -том ярусе помещены вершины, которые имеют входящие дуги из предыдущих ярусов, среди которых хотя бы одна дуга из $(k-1)$ -того яруса.

Количество вершин в ярусе определяет **ширину яруса**. Наибольшая ширина яруса определяет **ширину графа** в ярусно-параллельной форме. Количество ярусов определяет **высоту графа** в ярусно-параллельной форме.

Алгоритм приведения графа к ярусно-параллельной форме.

1. Составляется матрица смежности графа.
2. Матрица смежности просматривается в поисках нулевых столбцов. Вершины, которым соответствуют нулевые столбцы, помещаются в нулевой ярус.
3. Из матрицы смежности столбцы и строки, соответствующие вершинам нулевого яруса.
4. Повторяется п.2 данного алгоритма до тех пор, пока не будут охвачены все вершины.
5. По исходной матрице смежности восстанавливаются дуги между вершинами.

ПРИМЕР

Привести граф к ярусно-параллельной форме.

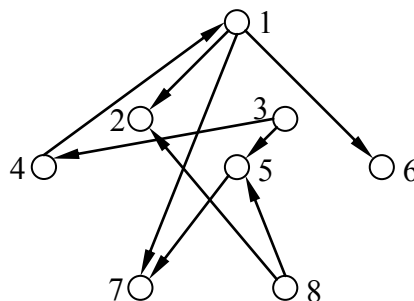


Рис. 3.8 Граф

1. Матрица смежности имеет вид:

	1	2	3	4	5	6	7	8
1		1				1	1	
2								
3				1	1			
4	1							
5							1	
6								
7								
8		1			1			

2. В нулевой ярус помещаются вершины 3 и 8.
3. Из матрицы смежности вычеркиваются строки и столбцы, соответствующие вершинам 3 и 8:

	1	2	4	5	6	7
1		1			1	1
2						
4	1					
5						1
6						
7						

4. В первый ярус помещаются вершины 4 и 5.
5. Из матрицы смежности вычеркиваются строки и столбцы, соответствующие вершинам 4 и 5:

	1	2	6	7
1		1	1	1
2				
6				
7				

6. Во второй ярус помещается вершина 1.
7. Из матрицы смежности вычеркиваются строка и столбец, соответствующие вершине 1:

	2	6	7
2			0
6			
7			

8. В третий (последний) ярус помещаются вершины 2,6 и 7.
- Таким образом, граф может быть представлен в ярусно-параллельной форме:

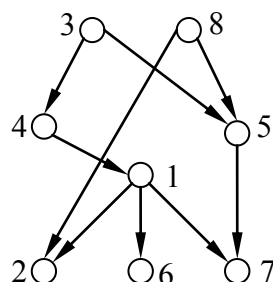


Рис. 3.9 Граф

Высота графа - 4 яруса; ширина -3.

Деревья и леса

Отделенными называются вершины, для которых не существует соединяющего эти вершины пути.

Неотделенными называются вершины, между которыми существует путь.

Подграфом графа $G = \langle \Gamma, X \rangle$ называется граф вида:

$$\begin{aligned} G_A &= \langle \Gamma_A, A \rangle \\ A &\subseteq X \quad \Gamma_A \subseteq \Gamma \end{aligned} \quad (3.19)$$

Компонентой связности называется подграф, порождаяемый множество неотделенных вершин.

ПРИМЕР

На рис. 3.10 представлен граф, имеющий две компоненты связности $\{1,2,3,4\}$ и $\{5,6\}$

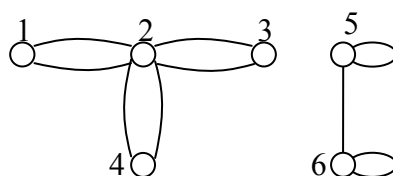


Рис. 3.10 Граф

Связанным графом называется граф, имеющий одну компоненту связности.

Дерево – связный неориентированный граф, в котором отсутствует цикл.

На рис. 3.11 представлены графы- деревья.



Рис. 3.11 Граф

Лесом называется граф, имеющий несколько компонентов связности, причем каждая из компонент является деревом.

Цикломатическим числом (γ) графа называется минимальное количество ребер, которое необходимо изъять из графа, чтобы он стал деревом. Цикломатическое число определяется по формуле:

$$\gamma = N - (k - 1), \tag{3.20}$$

где N – количество ребер,
 k – количество вершин.

Алгоритм получения дерева из графа

1. Выбирается любая вершина. Счетчик i принимаем равным 1 ($i=1$).
2. Если $i = k$, то дерево построено.
3. Если $i \neq k$, то выбирается любая незадействованная вершина и ребро, соединяющее данную вершину с любой из выбранных.
4. Счетчик увеличивается на единицу. ($i = i+1$).
5. Переход на пункт 2.

ПРИМЕР

Преобразовать граф, представленный на рис. 3.12 в дерево:

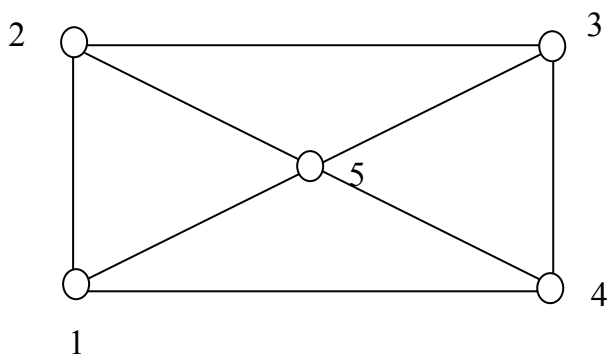


Рис. 3.12

1. Найдем цикломатическое число $\gamma = N - (k - 1) = 8 - (5 - 1) = 4$, т.е. необходимо изъять 4 ребра.
2. Выбираем вершину 5. $i = 1$
3. Присоединяем вершину 3.
4. $i = i + 1 = 2$
5. Присоединяем вершину 4.
6. $i = i + 1 = 3$
7. Присоединяем вершину 1.
8. $i = i + 1 = 4$
9. Присоединяем вершину 2.
10. $i = i + 1 = 5$
11. Все вершины охвачены. Дерево построено

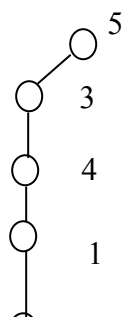


Рис. 3.13

ТЕОРИЯ АЛГОРИТМОВ

Алгоритм – это точное, понятное предписание о том, какие действия и в каком порядке должны быть выполнены, чтобы решить любую задачу из класса однотипных задач.

Алгоритм обладает следующими *характеристиками*:

1. **Дискретность.** Алгоритм – это процесс последовательного построения величин таким образом, что в начальный момент задается исходная конечная система величин, а в каждый следующий момент система величин получается по определенному закону из системы, имевшихся в предыдущий момент времени.

2. **Детерминированность.** Система величин, получаемых в какой-то не начальный момент, однозначно определяется системой величин, полученных в предшествующие моменты времени.

3. **Элементарность шагов алгоритмов.** Закон получения последующей системы величин из предшествующих должен быть простым.

4. **Массовость алгоритма.** Начальная система величин может выбираться из некоторых потенциально бесконечного множества. Иначе говоря, алгоритм должен обеспечивать решение некоторому множеству (классу) задач с различными параметрами (коэффициентами).

5. **Результативность алгоритма.** Последовательный процесс построения величин должен быть конечным и давать результат, того есть решение задачи.

Основная задача теории алгоритмов – это решение **проблемы алгоритмической разрешимости**, а не поиск правила (способа/метода) ее решения. Теория алгоритмов дает ответ на вопрос «Данная задача имеет решение?», и не отвечает на вопрос «Как решается данная задача?»

В рамках такого подхода к определению понятия алгоритма можно определить **три основных направления**:

1. Первое направление связано с уточнением понятия эффективно вычисляемой функции. В результате был выделен класс так называемых **рекурсивных функций**.

2. Второе направление связано с машинной математикой. Здесь сущность понятия алгоритма раскрывается путем рассмотрения процессов, осуществляемых в машине. Впервые это было сделано Тьюрингом, который предложил общую и вместе с тем самую простую концепцию вычислительной машины. Ее описание было дано Тьюрингом в 1937 г. А это направление в теории алгоритмов получило название - **машина Тьюринга**.

3. Третье направление связано с понятием нормальных алгоритмов, введенным и разработанным российским математиком А. А. Марковым. х Нормальные алгоритмы Маркова. Это направление получило название **нормальные алгоритмы Маркова**.

Рекурсивная функция

Основные понятия: элементарные функции, правила образования новых функций.

Простейшие функции:

1. **Функция сохранения нуля (нуль-функция)**

$$\forall x \in N \quad Z(x) = 0 \tag{4.1}$$

2. **Функция сдвига**
 $\forall x \in N \quad S(x) = x + 1$ (4.2)

3. **Функция-проекция**
 $\forall x \in N \quad I_k^n(x_1, x_2, \dots, x_k, \dots, x_n) = x_k$ (4.3)

Правила преобразования функций

1. Правило подстановки (суперпозиции)

Пусть даны функции:

$$g(y_1, y_2, \dots, y_n); \quad h_1(x_1, x_2, \dots, x_m); \quad h_2(x_1, x_2, \dots, x_m); \dots h_n(x_1, x_2, \dots, x_m) \quad \text{Тогда}$$

$$f(x_1, \dots, x_n) = g(h_1(x_1, \dots, x_m), \dots, h_n(x_1, \dots, x_m)) \quad (4.4)$$

где g и h являются или простейшими, или выведенными из простейших.

Правило вывода (4.4) означает, что функция f получена из функций g, h_1, \dots, h_n

правилом суперпозиции

ПРИМЕР

Функция $f(x) = k$ может быть получена путем применения k раз правила суперпозиции на основе функций $Z(x), S(x)$

$$f(x) = \underbrace{S(\dots S(Z(x)))}_k = k, \quad (4.5)$$

2. Правило примитивной рекурсии

Основывается на простейших или выведенных из простейших функциях g и h :

Пусть

$$g(x_1, x_2, \dots, x_n); \quad h(x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2})$$

Тогда новая функция $f(y, x_1, \dots, x_n)$ может быть выведена по правилу:

$$f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n)$$

$$f(x_1, \dots, x_n, y + 1) = h((x_1, \dots, x_n), f(x_1, \dots, x_n), y) \quad (4.6)$$

Следует отметить, что функция g зависит от n аргументов, функция f зависит от $n + 1$ аргументов, функция h зависит от $n + 2$ аргументов. Иначе говоря, правило примитивной рекурсии позволяет получить $n + 1$ -местную функцию из n -местной и $n + 2$ -местной функций.

ПРИМЕР

Пусть некоторая функция $f(x, y)$ задана правилом рекурсии

$$f(x, 0) = I_1^2(x, y) = x$$

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

Нетрудно заметить, что функция $g(x) = I_1^2(x, y) = x$, функция

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

Вычислим значение функции $f(x, y)$ при $x = 2, \quad y = 5$.

$$f(2,0) = 2$$

$$f(2,1) = f(x,0) + 1 = 2 + 1 = 3$$

$$f(2,2) = f(x,1) + 1 = 3 + 1 = 4$$

$$f(2,3) = f(x,2) + 1 = 4 + 1 = 5$$

$$f(2,4) = f(x,3) + 1 = 5 + 1 = 6$$

$$f(2,5) = f(x,4) + 1 = 6 + 1 = 7$$

Нетрудно заметить, что функция $f(x, y)$ выполняет сложение двух чисел x и y .

3. μ -оператор (оператор нахождения наименьшего корня y)

Оператор $\mu y(f(x, y))$ определяет наименьшее значение y , при котором $f(x, y) = 0$ при фиксированном значении. Принято обозначение

$$\varphi(x) = \mu y[f(x, y) = 0] \quad (4.7)$$

(Читается: «наименьшее y такое, что $f(x, y) = 0$ »). Аналогично определяется функция многих переменных :

$$\varphi(x_1, x_2, \dots, x_n) = \mu y[f(x_1, x_2, \dots, x_n, y) = 0] \quad (4.8)$$

Для вычисления функции φ существует следующий алгоритм:

1. Вычисляется $f(x_1, x_2, \dots, x_n, 0)$. Если это значение функции f равно нулю, то $\varphi(x_1, x_2, \dots, x_n) = 0$. Если $f(x_1, x_2, \dots, x_n, 0) \neq 0$, то осуществляется переход к следующему шагу.
2. Вычисляется $f(x_1, x_2, \dots, x_n, 1)$. Если это значение функции f равно нулю, то $\varphi(x_1, x_2, \dots, x_n) = 1$. Если $f(x_1, x_2, \dots, x_n, 1) \neq 0$, то осуществляется переход к следующему шагу. И т. д.

Если окажется, что для всех y функция $f(x_1, x_2, \dots, x_n, y) \neq 0$, то функция $\varphi(x_1, x_2, \dots, x_n)$ считается неопределенной.

ПРИМЕР

Дана функция $f(x, y) = x - y - 2$. Необходимо определить $\varphi(x) = \mu y[f(x, y) = 0]$ при $x = 7$

$$y = 0 \quad 7 - 0 - 2 = 5 \neq 0$$

$$y = 1 \quad 7 - 1 - 2 = 4 \neq 0$$

$$y = 2 \quad 7 - 2 - 2 = 3 \neq 0$$

$$y = 3 \quad 7 - 3 - 2 = 2 \neq 0$$

$$y = 4 \quad 7 - 4 - 2 = 1 \neq 0$$

$$y = 5 \quad 7 - 5 - 2 = 0$$

Таким образом, $\varphi(x = 7) = 5$

Функция $f(x_1, x_2, \dots, x_n)$ называется **частично рекурсивной**, если она получена из простейших функций за конечное число шагов на основе правил подстановки, примитивной рекурсии или μ -оператора.

Функция $f(x_1, x_2, \dots, x_n)$ называется **примитивно рекурсивной**, если она может быть получена из простейших функций за конечное число шагов на основе правил подстановки, примитивной рекурсии.

Функция называется **общерекурсивной**, если она частично рекурсивная и всюдуопределенная.

Тезис А. Черча. Если функция является общерекурсивной, то она **выполнима**, т.е. имеет алгоритм решения.

Машина Тьюринга

Если для решения некоторой массовой проблемы известен алгоритм, то для его реализации необходимо лишь четкое выполнение предписаний этого алгоритма. Автоматизм, необходимый при реализации алгоритма, приводит к мысли о передаче функции человека, реализующий алгоритм, машине. Идею такой машины предложил в 1937 году английский математик А. Тьюринг.

Машина Тьюринга включает в себя:

1. **Внешний алфавит** - конечное множество символов $A = \{a_0, a_1, a_2, \dots, a_n\}$. В этом алфавите в виде слова кодируется та информация, которая подается в машину. Машина перерабатывает информацию, поданную в виде слова, в новое слово. Обычно символ **Внешний алфавит** - конечное множество символов a_0 обозначает пробел.
2. **Внутренний алфавит** - конечное множество символов $Q = \{q_0, q_1, q_2, \dots, q_m\}$. Для любой машины число состояний фиксировано. Два состояния имеют особое назначение q_1 - начальное состояние машины, q_0 - заключительное состояние (стоп-состояние).
3. **Операторы перемещения** $T = \{Л, П, Н\}$. Л, П, Н – это символы сдвига «влево», «вправо» и «на месте».
4. **Бесконечная лента** Бесконечная лента характеризует память машины. Она разбита на клеточки. В каждую клеточку может быть записан только один символ из внешнего алфавита.
5. **Управляющая головка.** Управляющая головка (УГ) передвигается вдоль ленты и может останавливаться напротив какой-либо клетки, т. е. считывать символ
6. **Управляющая головка.** Управляющая головка (УГ) передвигается вдоль ленты и может останавливаться напротив какой-либо клетки, т. е. считывать символ.

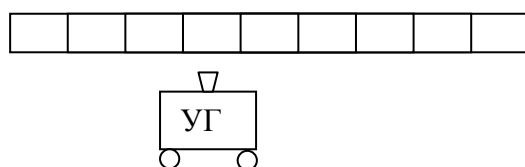


Рис. 4.1. Функциональная схема машины Тьюринга.

7. **Программа машины Тьюринга (P)** - совокупность всех команд, Программа представляется в виде таблицы и называется **Тьюринговой функциональной схемой**.

	a_0	a_1	a_2
q_1	$a_0 П q_1$	$a_1 П q_1$	$a_2 Л q_2$
q_2	$a_1 П q_2$	$a_2 Н q_0$	$a_0 Н q_0$

Таким образом, машина Тьюринга может быть представлена в виде четверки:

$$MT = \langle A, Q, T, P \rangle \quad (4.9)$$

Работа машины Тьюринга:

Информация, хранящаяся на ленте, является набором символов из внешнего алфавита. Начальное состояние управляющей головки характеризуется символом внутреннего алфавита q_1 . Работа машины складывается из тактов. В течение любого такта машина Тьюринга осуществляет следующие действия: машина Тьюринга находится во внутреннем состоянии q_i , считывает входной символ a_j и по таблице работы совершает операцию сдвига t_l , переходя в состояние q_k , при этом входное слово a_j заменяется на a_m :

$$a_j q_i \rightarrow a_m t_l q_k$$

Если в результате операции машина перейдет в состояние q_0 , то работа машины останавливается. Если состояние q_0 недостижимо, то значит по данному входному слову машина Тьюринга не достигает конечного состояния и алгоритма для данного входного слова не существует.

ПРИМЕР

Построить машину Тьюринга, которая будет стирать последнюю единицу в последовательности единиц.

Внешний алфавит - $A = \{a_0, 1\}$. Внутренний алфавит - $Q = \{q_0, q_1, q_2\}$, при этом состояние q_1 сохраняется до тех пор, пока не будет найден конец последовательности единиц, состояние q_2 - стирание последней единицы. При этом следует заметить, что ситуация $a_0 q_2$ в работе машины Тьюринга невозможна, поэтому соответствующая клеточка доопределена произвольно, например $a_0 П q_2$. Начальное состояние q_1 на начале последовательности единиц. Рабочая программа машины Тьюринга имеет вид:

	1	a_0
q_1	1 П q_1	$a_0 Л q_2$
q_2	$a_0 Н q_0$	$a_0 П q_2$

Проверим работоспособность машины Тьюринга:

1. $a_0 \overset{11}{a_0}$
 q_1
2. $a_0 \overset{11}{a_0}$
 q_1
3. $a_0 \overset{11}{a_0}$
 q_1
4. $a_0 \overset{11}{a_0}$
 q_2
5. $a_0 \overset{10}{a_0}$
 q_0

Тезис А. Черча. Если функция *выполнима*, то она всегда может быть представлена в виде машины Тьюринга.

Нормальные алгоритмы Маркова

Нормальный алгоритм Маркова представляет собой систему подстановок

$$P \rightarrow (\bullet) Q, \quad (4.10)$$

где P, Q - слова из символов входного алфавита A

$\rightarrow (\bullet)$ - оператор подстановки, при этом :

\rightarrow -простая подстановка

$\rightarrow \bullet$ - заключительная подстановка.

Слово z считается **включенным в слово u** , если u может быть представлено как:

$$x_1 z x_2, \quad (4.11)$$

где x_1, x_2 - любые символы входного алфавита. A

Работа нормального алгоритма Маркова:

Исходное слово просматривается слева направо с целью выявления вхождения первого правила подстановки. Как только находится первое вхождение первого правила подстановки, оно заменяется по этому правилу и исходное слово снова просматривается с первого символа по первому правилу подстановки.

После того, как первое правило больше не встречается в данном слове, аналогично применяется второе правило подстановки.

Работа алгоритма заканчивается тогда, когда ни одна из подстановок не применима, либо использована заключительная подстановка.

ПРИМЕР

Построить нормальный алгоритм Маркова, стирающий последовательность единиц.

Нормальный алгоритм Маркова для данной задачи представляет собой две подстановки :

1. $11 \rightarrow 1$

2. $1a_0 \rightarrow \bullet a_0 a_0$

Первая подстановка стирает все единицы до последней. Вторая (заключительная) подстановка заменяет последнюю единицу пробелом a_0 .

Тезис А. Черча. Если функция *выполнима*, то она может быть представлена в виде нормального алгоритма Маркова.

Заключительный тезис А. Черча. Если функция *выполнима*, то она может быть представлена в виде либо общерекурсивной функции, либо машины Тьюринга, либо в виде нормального алгоритма Маркова.

ТЕОРИЯ АВТОМАТОВ

Автомат – дискретный преобразователь информации, который на основе входных сигналов, поступающих в дискретные моменты времени, и с учетом своего состояния вырабатывает выходные сигналы и изменяет свое состояние.

В данном разделе рассматриваются абстрактные автоматы, т.е. некоторая математическая модель. Вопросы практической реализации не рассматриваются. В связи с этим при построении автоматов будем иметь в виду, что:

1. Автомат функционирует в абстрактном времени.
2. Все переходы происходят мгновенно.

Автомат представляет собой кортеж 6 –го порядка:

$$\alpha = \langle X, Y, A, f(t), \varphi(t), a_0 \rangle, \quad (5.1)$$

где X – множество входных сигналов (входной алфавит),
 Y – множество выходных сигналов (выходной алфавит),
 A – множество внутренних состояний,
 $f(t)$ – функция перехода,
 $\varphi(t)$ – функция выхода,
 $a_0 \in A$ – начальное состояние автомата.

Законы функционирования автоматов.

В зависимости от законов функционирования различают 3 вида автоматов:

1. Автоматы первого рода, или автоматы Мили:

$$\begin{aligned} a(t) &= f(a(t-1), x(t)) \\ y(t) &= \varphi(a(t-1), x(t)) \end{aligned} \quad (5.2)$$

2. Автоматы второго рода

$$\begin{aligned} a(t) &= f(a(t-1), x(t)) \\ y(t) &= \varphi(a(t), x(t)) \end{aligned} \quad (5.3)$$

3. Правильные автоматы второго рода, или автоматы Мура:

$$\begin{aligned} a(t) &= f(a(t-1), x(t)) \\ y(t) &= \varphi(a(t)) \end{aligned} \quad (5.4)$$

На практике наибольшее распространение получили автоматы Мили и автоматы Мура.

Задание автоматов

Автоматы могут быть заданы следующими способами:

1. В виде графа

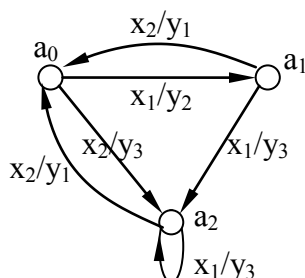


РИС. 5.1. Автомат Мили

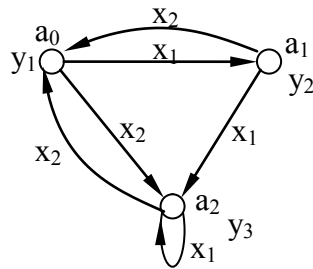


РИС. 5.2. Автомат Мура.

При построении автомата Мили каждая дуга, соединяющая вершины a_i и a_j , имеет обозначение x_k / y_m . Это означает следующее: находясь в состоянии a_i , автомат, обрабатывая входной сигнал x_k , выдает выходной сигнал y_m и переходит в состояние a_j .

Так как в автомате Мура выходной сигнал y_m зависит только от текущего состояния a_j , то каждая дуга, соединяющая вершины a_i и a_j , имеет обозначение x_k .

2. В виде таблиц перехода и выхода (автомат Мили); отмеченной таблицы перехода (автомат Мура).

Автомат Мили описывается с помощью двух таблиц: таблицы перехода и таблицы выхода:

Таблица переходов (ТП)

	a_0	a_1	a_2
x_1	a_1	a_2	a_2
x_2	a_2	a_0	a_0

Таблица выходов (ТВ)

	a_0	a_1	a_2
x_1	y_2	y_3	y_3
x_2	y_3	y_1	y_1

Автомат Мура описывается с помощью отмеченной таблицы перехода:

Таблица переходов (ТП)

	y_1	y_2	y_3
	a_0	a_1	a_2
x_1	a_1	a_2	a_2
x_2	a_2	a_0	a_0

ПРИМЕР.

Синтезировать автомат, на вход которого подаются монеты номинальной стоимостью 1, 2 и 3 копейки, а на выходе автомат выдает билет, если сумма набранных монет составляет 3 копейки, если сумма меньше 3 копеек, то автомат ничего не выдает, если сумма больше 3 копеек, то автомат возвращает деньги.

Определим входной, выходной алфавиты и множество внутренних состояний:

- входной алфавит $X = \{1, 2, 3\}$ - монеты номинальной стоимостью 1, 2 и 3 копейки
- выходной алфавит $Y = \{H, Б, В\}$ - на выходе возможны выходные символы: H - ничего; $Б$ - билет; $В$ - возврат.
- множество внутренних состояний $A = \{a_0, a_1, a_2, a_3\}$,

где a_0 - начальное состояние автомата « в автомате ничего нет»;

a_1 - «в автомате 1 копейка»;

a_1 - «в автомате 1 копейка»;

a_2 - «в автомате 2 копейки»;

a_3 - «в автомате 3 копейки».

Граф автомата имеет вид:

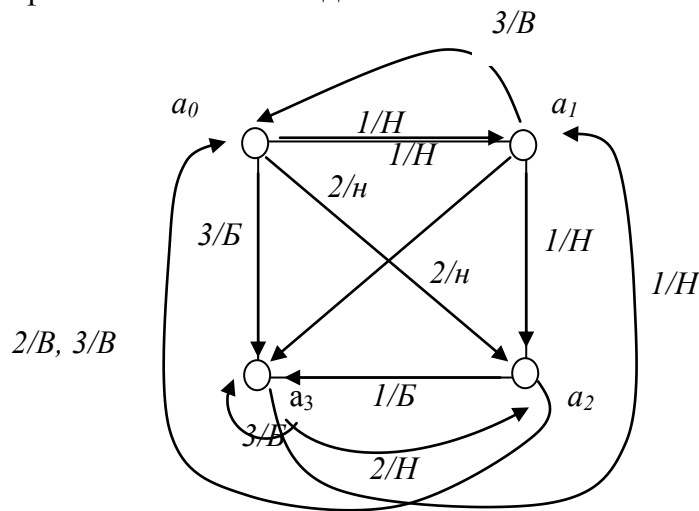


РИС. 5.3. Автомат Мили

Таблицы перехода и выхода представлены в виде:

Таблица переходов (ТП)

	a_0	a_1	a_2	a_3
1	a_1	a_2	a_3	a_1
2	a_2	a_3	a_0	a_2
3	a_3	a_0	a_0	a_3

Таблица выходов (ТВ)

	a_0	a_1	a_2	a_3
1	Н	Н	Б	Н
2	Н	Б	В	Н
3	Б	В	В	Б

Неопределенным состоянием называется несуществующее состояние.

Частичным автоматом называется автомат, в котором некоторые состояния в таблице перехода не определены. Для дальнейшего исследования неопределенное состояние некоторым образом доопределяют.

Минимизация автоматов

Входным словом называется совокупность сигналов, поступающих на вход.

Выходным словом называются совокупность сигналов на выходе.

Два автомата называются **эквивалентными**, если они имеют одинаковый входной и выходной алфавит, и на одинаковые входные слова выдают одинаковые выходные слова.

Два состояния **одноэквивалентными**, если на одинаковое входное слово выдается одинаковый выходной сигнал.

Два состояния **k-эквивалентными**, если на одинаковое входное слово длиной в k-единиц выдается одинаковый выходной сигнал длиной в k-единиц.

Эквивалентными состояниями называются k-эквивалентные состояния для любых k.

Эквивалентные состояния объединяются в **класс эквивалентности**.

Минимальный автомат – это автомат, состоящий из наименьшего числа состояний, каждое из которых является классом эквивалентности исходного автомата.

Алгоритм минимизации автомата Мили

1. По таблице выхода находятся состояния с одинаковыми выходными сигналами. Данные состояния объединяются в класс одноэквивалентных состояний. Проводится перекодировка.

2. По таблице перехода определяются классы двухэквивалентных состояний: для любого класса выделяется состояние, которое на одинаковый входной сигнал переходит в одинаковое состояние. Объединяем двухэквивалентные состояния в классы двухэквивалентных состояний. Проводится перекодировка.

3. Алгоритм выполняется, пока в классах k-эквивалентных состояний не находятся одинаковые состояния.

4. Вводятся новые состояния, соответствующие классам эквивалентных состояний.

5. С учетом новых состояний переписываются таблицы перехода и выхода.

ПРИМЕР

Пусть задан автомат Мили

Таблица выходов

	1	2	3	4	5	6	7	8	9
x_1	1	0	1	0	1	0	1	1	0
x_2	0	1	0	1	0	1	0	0	1
x_3	0	1	0	1	0	1	0	0	1

Таблица переходов

	1	2	3	4	5	6	7	8	9
x_1	2	1	2	3	6	8	6	4	7
x_2	2	4	2	2	4	9	2	4	9
x_3	5	4	5	2	3	6	8	7	7

Определяем класс одноэквивалентных состояний по таблице выхода

Таблица выходов

	1	2	3	4	5	6	7	8	9
x_1	1	0	1	0	1	0	1	1	0
x_2	0	1	0	1	0	1	0	0	1
x_3	0	1	0	1	0	1	0	0	1
	a	b	a	b	a	b	a		b

Выделяются два класса одноэквивалентных состояний $a=\{1,3,5,7,8\}$ и $b=\{2,4,6,9\}$.
Перегруппируем таблицу перехода по классам одноэквивалентных состояний

Таблица переходов

	a					b			
	1	3	5	7	8	2	4	6	9
x_1	2	2	6	6	4	1	3	8	7
x_2	2	2	4	2	4	4	2	9	9
x_3	5	5	3	8	7	4	2	6	7

Перекодируем состояния по полученным классам

Таблица переходов

	a					b			
	1	3	5	7	8	2	4	6	9
x_1	2/ b	2/ b	6/ b	6/ b	4/ b	1/ a	3/ a	8/ a	7/ a
x_2	2/ b	2/ b	4/ b	2/ b	4/ b	4/ b	2/ b	9/ b	9/ b
x_3	5/ a	5/ a	3/ a	8/ a	7/ a	4/ b	2/ b	6/ b	7/ a

Выделяем внутри каждого из классов одинаковые состояния, тем самым определяя классы двухэквивалентных состояний

Таблица переходов

	a					b			
	1	3	5	7	8	2	4	6	9
x_1	2/ b	2/ b	6/ b	6/ b	4/ b	1/ a	3/ a	8/ a	7/ a
x_2	2/ b	2/ b	4/ b	2/ b	4/ b	4/ b	2/ b	9/ b	9/ b
x_3	5/ a	5/ a	3/ a	8/ a	7/ a	4/ b	2/ b	6/ b	7/ a
	a					b			c

Определим новые классы двухэквивалентных состояний $a=\{1,3,5,7,8\}$, $b=\{2,4,6\}$, $c=\{9\}$, перекодируем по новым состояниям и выделим классы трехэквивалентных состояний

Таблица переходов

	a					b			c
	1	3	5	7	8	2	4	6	9

x_1	2/в	2/в	6/в	6/в	4/в	1/а	3/а	8/а	7/а
x_2	2/в	2/в	4/в	2/в	4/в	4/в	2/в	9/с	9/с
x_3	5/а	5/а	3/а	8/а	7/а	4/в	2/в	6/в	7/а
	а					в		с	д

Классы трехэквивалентных состояний $a=\{1,3,5,7,8\}$, $b=\{2,4\}$, $c=\{6\}$, $d=\{9\}$ перекодируем по новым состояниям и выделим классы четырехэквивалентных состояний

Таблица переходов

	а					в		с	д
	1	3	5	7	8	2	4	6	9
x_1	2/в	2/в	6/с	6/с	4/в	1/а	3/а	8/а	7/а
x_2	2/в	2/в	4/в	2/в	4/в	4/в	2/в	9/д	9/д
x_3	5/а	5/а	3/а	8/а	7/а	4/в	2/в	6/с	7/а
	а		в		а	с		д	е

Перегруппируем таблицу перехода по новым классам $a=\{1,3,8\}$, $b=\{5,7\}$, $c=\{2,4\}$, $d=\{6\}$, $e=\{9\}$, перекодируем по новым состояниям.

Таблица переходов

	а			в		с		д	е
	1	3	8	5	7	2	4	6	9
x_1	2/с	2/с	4/с	6/д	6/д	1/а	3/а	8/а	7/б
x_2	2/с	2/с	4/с	4/с	2/с	4/с	2/с	9/е	9/е
x_3	5/в	5/в	7/в	3/а	8/а	4/с	2/с	6/д	7/б
	а			в		с		д	е

Так как внутри из каждого класса дальнейшее разбиение на классы не осуществляется, это означает, что найдены классы эквивалентных состояний: $a=\{1,3,8\}$, $b=\{5,7\}$, $c=\{2,4\}$, $d=\{6\}$, $e=\{9\}$.

Минимизированный автомат Мили в новых состояниях имеет вид

Таблица переходов

	а	б	с	д	е
x_1	с	д	а	а	б
x_2	с	с	с	е	е
x_3	в	с	с	д	б

Таблица выходов

	а	б	с	д	е
x_1	1	1	0	0	0

x_2	0	0	1	0	0
x_3	0	0	1	1	1

Особенности минимизации автомата Мура.

:

Автомат Мура минимизируется аналогично минимизации автомата Мили за исключением первого шага. Выделение класса одноэквивалентных состояний осуществляется по строке выходов отмеченной таблицы переходов автомата Мура.

Минимизация частичных автоматов.

Для того, чтобы провести минимизацию частичных автоматов неопределенное состояние доопределяется самостоятельно. Далее минимизация автоматов осуществляется по вышеизложенному алгоритму.

Переход от автомата Мили к автомату Мура

Автоматы Мили и автоматы Мура отличаются функцией выхода.

Автомат Мили:

$$y(t) = \varphi(a(t-1), x(t)) \quad (5.5)$$

Автомат Мура:

$$y(t) = \varphi(a(t)) \quad (5.6)$$

То есть произвольному состоянию автомата Мили a_i и входному сигналу x_j соответствует состояние b_{ij} автомата Мура:

$$a_i x_j \rightarrow b_{ij} \quad (5.7)$$

При этом начальные состояния автоматов Мили и Мура совпадают:

$$a_0 = b_0 \quad (5.8)$$

Учитывая вышеизложенное, можно перекодировать таблицу перехода автомата Мили и составить отмеченную таблицу переходов автомата Мура.

ПРИМЕР

Пусть задан автомат Мили

Таблица переходов (ТП)

	a_0	a_1	a_2
x_1	a_1	a_2	a_2
x_2	a_2	a_0	a_0

Таблица выходов (ТВ)

	a_0	a_1	a_2
x_1	y_2	y_3	y_3
x_2	y_3	y_1	y_1

Перекодируем матрицу перехода автомата Мили:

	$a_0 = b_0$	a_1	a_2
--	-------------	-------	-------

x_1	a_1/b_{01}	a_2/b_{11}	a_2/b_{21}
x_2	a_2/b_{02}	a_0/b_{12}	a_0/b_{22}

Составляем таблицу перехода автомата Мура.

	b_0	b_{01}	b_{02}	b_{11}	b_{12}	b_{21}	b_{22}
x_1	b_{01}	b_{11}	b_{21}	b_{21}	b_{01}	b_{21}	b_{01}
x_2	b_{02}	b_{12}	b_{22}	b_{22}	b_{02}	b_{22}	b_{02}

При составлении таблицы перехода автомата Мили рассуждаем следующим образом: состояние автомата Мура b_{01} соответствует состоянию автомата Мили a_1 , следовательно, столбец состояния b_{01} автомата Мура совпадает со столбцом состояния a_1 автомата Мили.

Так как в автомате Мура произвольному состоянию b_{ij} соответствует некоторый выходной сигнал, то строка выхода отмеченной таблицы перехода автомата Мура однозначно определяется таблицей выхода автомата Мили (состоянию b_{01} соответствует выходной сигнал y_2 ; $b_{02} - y_3$)

		y_2	y_3	y_3	y_1	y_3	y_1
	b_0	b_{01}	b_{02}	b_{11}	b_{12}	b_{21}	b_{22}
x_1	b_{01}	b_{11}	b_{21}	b_{21}	b_{01}	b_{21}	b_{01}
x_2	b_{02}	b_{12}	b_{22}	b_{22}	b_{02}	b_{22}	b_{02}

Выходной сигнал, соответствующий состоянию b_0 , выбирается произвольно.

	y_2	y_2	y_3	y_3	y_1	y_3	y_1
	b_0	b_{01}	b_{02}	b_{11}	b_{12}	b_{21}	b_{22}
x_1	b_{01}	b_{11}	b_{21}	b_{21}	b_{01}	b_{21}	b_{01}
x_2	b_{02}	b_{12}	b_{22}	b_{22}	b_{02}	b_{22}	b_{02}

Если автомат Мили содержит m -состояний и n входных символов, то количество состояний автомата Мура определяется по формуле:

$$k = m \times n + 1 \quad (5.9)$$

Переход от автомата Мура к автомату Мили

Переход от автомата Мура к автомату Мили заключается в построении таблицы выходов. Построение состоит в подстановке выходных сигналов, отмечающих состояния в отмеченной таблице переходов, вместо состояний, в которые автомат переходит. Тем самым, если говорить в терминах графов, выходные сигналы от состояний переносятся на дуги, которые в эти состояния заходят.

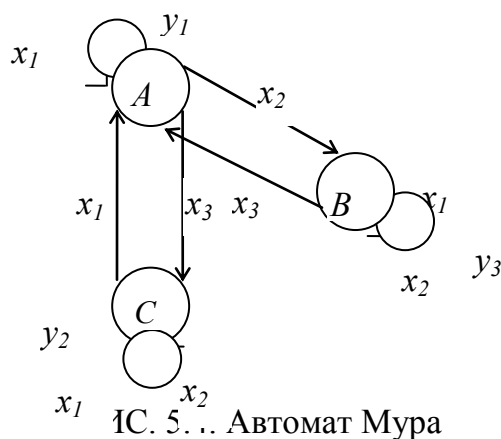
А таблица переходов автомата Мили получается из отмеченной таблицы переходов автомата Мура отбрасыванием строки выходов.

ПРИМЕР

Пусть задан автомат Мура в виде отмеченной таблицы перехода

	y_1	y_3	y_2
	A	B	C
x_1	A	B	A
x_2	B	B	C
x_3	C	A	C

Данный автомат может быть представлен в виде графа:



ИС. 5... Автомат Мура

Автомат Мили будет иметь вид:

- в виде таблиц перехода и выхода

Таблица переходов

	y_1	y_3	y_2
	A	B	C
x_1	A	B	A
x_2	B	B	C
x_3	C	A	C

Таблица выходов

	A	B	C
x_1	y_1	y_3	y_1
x_2	y_3	y_3	y_2
x_3	y_2	y_1	y_2

- в виде графа

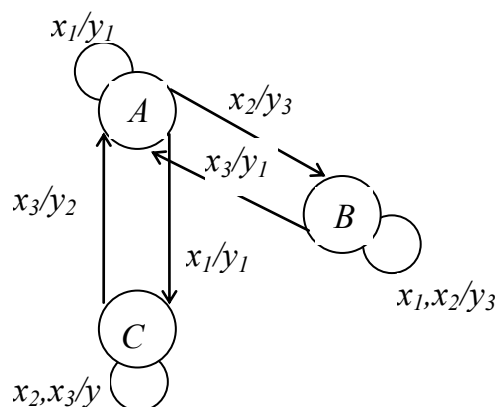


РИС. 5.5. Автомат Мили

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Работа с учебником.

Для обеспечения максимально возможного усвоения материала и с учётом индивидуальных особенностей студента, можно предложить следующие приёмы обработки информации учебника:

- конспектирование;
- составление плана учебного текста;
- тезирование;
- аннотирование;
- составление тематического тезауруса;
- выделение проблемы и нахождение путей её решения;
- самостоятельная постановка проблемы и нахождение в тексте путей её решения;
- определение алгоритма практических действий (план, схема).

В качестве форм и методов контроля внеаудиторной самостоятельной работы студентов могут быть использованы обмен информационными файлами, семинарские занятия, зачеты, тестирование, самоотчеты, контрольные работы, защита творческих работ и электронных презентаций и др.

Контроль результатов внеаудиторной самостоятельной работы студентов может осуществляться в пределах времени, отведенного на обязательные учебные занятия по дисциплине или в специально отведенное время (зачет, экзамен).