

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №1

з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-114
Ярка Ірина
Викладач:
Мельникова Н.І.

Львів – 2019 р.

Моделювання основних логічних операцій

Мета роботи: Ознайомитись на практиці із основними поняттями математичної логіки, навчитись будувати складні висловлювання за допомогою логічних операцій та знаходити їхні істинні значення таблицями істинності, використовувати закони алгебри логіки, освоїти методи доведень.

Теоретичні відомості

Просте висловлювання (атомарна формула, атом) – це розповідне речення, про яке можна сказати, що воно *істинне* (Т або 1) або *хибне* (F або 0), але не те й інше водночас.

Складне висловлювання – це висловлювання, побудоване з простих за допомогою *логічних операцій (логічних зв'язок)*. Найчастіше вживаними операціями є 6: **заперечення** (читають «не», позначають \neg , $-$), **кон'юнкція** (читають «і», позначають \wedge), **диз'юнкція** (читають «або», позначають \vee), **імплікація** (читають «якщо ..., то», позначають \Rightarrow), **альтернативне «або»** (читають «додавання за модулем 2», позначають \oplus), **еквівалентність** (читають «тоді і лише тоді», позначають \Leftrightarrow).

Запереченням довільного висловлювання P називають таке висловлювання $\neg P$, істинне значення якого строго протилежне значенню P .

Кон'юнкцією або **логічним множенням** двох висловлювань P та Q називають складне висловлювання $P \wedge Q$, яке набуває істинного значення тільки в тому випадку, коли істинні *обидві* його складові.

Диз'юнкцією або **логічним додаванням** двох висловлювань P та Q називають складне висловлювання $P \vee Q$, яке набуває істинного значення в тому випадку, коли істинною є *хоча б одна* його складова.

Імплікацією двох висловлювань P та Q називають умовне висловлювання «якщо P , то Q » ($P \Rightarrow Q$), яке прийнято вважати *хибним* тільки в тому випадку, коли *передумова (антецедент) P істинна, а висновок (консеквент) Q хибний*.

Еквіваленцією двох висловлювань P та Q називають складне висловлювання $P \Leftrightarrow Q$, яке набуває істинного значення тоді і лише тоді, коли P та Q мають *однакові* логічні значення.

Тавтологія – формула, що виконується у всіх інтерпретаціях (тотожно істинна формула).

Протиріччя – формула, що не виконується у жодній інтерпретації (тотожно хибна формула).

A	B
Закони асоціативності	
$(P \vee Q) \vee R = P \vee (Q \vee R)$	$(P \wedge Q) \wedge R = P \wedge (Q \wedge R)$
Закони комутативності	
$P \vee Q = Q \vee P$	$P \wedge Q = Q \wedge P$
Закони ідемпотентності	
$P \vee P = P$	$P \wedge P = P$
Закони дистрибутивності	
$P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$	$P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$
Закони доповнення	
закон виключення третього: $P \vee (\overline{P}) = T$	закон протиріччя: $P \wedge (\overline{P}) = F$
закон подвійного заперечення $\overline{\overline{P}} = P$	
Закони де Моргана	
$\overline{(P \vee Q)} = \overline{P} \wedge \overline{Q}$	$\overline{(P \wedge Q)} = \overline{P} \vee \overline{Q}$
Закони поглинання	
$(P \vee Q) \wedge P = P$	$(P \wedge Q) \vee P = P$
Співвідношення для сталих (закони тотожності та домінування)	

Предикат – це твердження, яке містить змінні та приймає значення істини чи фальші залежно від значень змінних; ***n*-місний предикат** – це предикат, що містить *n* змінних x_1, \dots, x_n .

Квантор - логічний оператор, що перетворює будь-який предикат на предикат меншої місності, зв'язуючи деякі змінні початкового предиката. Вживаються два квантори: узагальнення (універсальний) (позначається \forall) та приналежності (екзистенціальний) (позначається \exists).

Випереджена нормальна форма – формула, записана у вигляді $Q_1x_1Q_2x_2\dots Q_nx_nM$, де кожне Q_ix_i ($i = 1, 2, \dots, n$) – це $\forall x_i$ або $\exists x_i$, а формула M не містить кванторів. Вираз $Q_1x_1\dots Q_nx_n$ називають префіксом, а M – *матрицею формули*, записаної у випередженій нормальній формі.

- **Варіант 15**

1. Формалізувати речення

Якщо не можеш зробити якісно роботу, то вважай, що тобі не запропонують вдалу вакансію.

Нехай q = зробити роботу якісно, p = запропонують вдалу вакансію, отже:

$$\neg q \Rightarrow \neg p$$

2. Побудувати таблицю істинності для висловлювань:

$$(x \wedge (y \wedge z)) \Rightarrow ((x \vee y) \vee z)$$

№	x	y	z	$y \wedge z$	$x \vee y$	$x \wedge (y \wedge z)$	$(x \vee y) \vee z$	$(x \wedge (y \wedge z)) \Rightarrow ((x \vee y) \vee z)$
0	1	1	1	1	1	1	1	1
1	0	1	1	1	1	0	1	1
2	1	0	1	0	1	0	1	1
3	1	1	0	0	1	0	1	1
4	1	0	0	0	1	0	1	1
5	0	0	1	0	0	0	1	1
6	0	1	0	0	1	0	1	1
7	0	0	0	0	0	0	0	1

3. Побудовою таблиць істинності вияснити, чи висловлювання є тавтологією або протиріччям:

$$\left(\overline{(p \wedge q)} \vee (\overline{q} \wedge r) \right) \vee \overline{(p \rightarrow r)}$$

1. $\overline{(p \wedge q)} = \neg p \vee \neg q$

2. $\overline{(p \rightarrow r)} = \neg p \wedge \neg r$

№	p	$\neg p$	q	$\neg q$	r	$\neg r$	$\neg p \wedge \neg q$	$\neg q \wedge r$	$(\neg p \vee \neg q) \vee (\neg q \wedge r)$	$\neg p \vee \neg r$	$((\neg p \vee \neg q) \vee (\neg q \wedge r)) \vee (\neg p \vee \neg r)$
0	1	0	1	0	1	0	0	0	0	0	0
1	0	1	1	0	1	0	1	0	1	0	1
2	1	0	0	1	1	0	1	1	1	0	1
3	1	0	1	0	0	1	0	0	0	0	0
4	1	0	0	1	0	1	1	0	1	0	1
5	0	1	0	1	1	0	1	1	1	0	1
6	0	1	1	0	0	1	1	0	1	1	1
7	0	1	0	1	0	1	1	0	1	1	1

Висловлювання не є ні тавтологією, ні протиріччям.

4. За означенням без побудови таблиць істинності та виконання еквівалентних перетворень перевірити, чи є тавтологією висловлювання:

$$(((\overline{p \rightarrow q}) \rightarrow p) \wedge ((\overline{p \rightarrow q}) \rightarrow r)) \rightarrow (p \rightarrow q)$$

Спробуємо довести від супротивного
Нехай:

$$(((\neg p \rightarrow \neg q) \rightarrow p) \wedge (\neg(p \rightarrow q) \rightarrow r)) = T;$$

$$(p \rightarrow q) = F, \text{ тоді}$$

$$((T \rightarrow p) \wedge (T \rightarrow r)) = T$$

$$F \rightarrow p = T$$

$$T \rightarrow r = T$$

Отже, висловлювання не є ні тавтологією, ні протиріччям. Для підтвердження цього побудуємо таблицю істинності:

$$*5 = (((p \rightarrow q) \rightarrow p) \wedge (\neg(p \rightarrow q) \rightarrow r))$$

$$7 = (((p \rightarrow q) \rightarrow p) \wedge (\neg(p \rightarrow q) \rightarrow r)) \rightarrow (p \rightarrow q)$$

№	p	q	r	$\neg p$	$\neg q$	$\neg p \rightarrow \neg q$	$(\neg p \rightarrow \neg q) \rightarrow p$	$\neg(p \rightarrow q)$	$\neg(p \rightarrow q) \rightarrow r$	5	$p \rightarrow q$	7
0	1	1	1	0	0	1	1	0	1	1	1	1
1	1	1	0	0	0	1	1	0	1	1	1	1
2	1	0	1	0	1	1	1	1	1	1	0	0
3	0	1	1	1	0	0	1	0	1	1	1	1
4	1	0	0	0	1	1	1	1	0	0	0	1
5	0	1	0	1	0	0	1	0	1	1	1	1
6	0	0	1	1	1	1	0	0	1	0	1	1
7	0	0	0	1	1	1	0	0	1	0	1	1

5. Довести, що формули еквівалентні:

$$(\neg q \wedge r) \rightarrow p \text{ та } p \rightarrow (q \wedge r)$$

$$((\neg q \wedge r) \rightarrow p) \leftrightarrow (p \rightarrow (q \wedge r))$$

№	p	q	$\neg q$	r	$\neg q \wedge r$	$(\neg q \wedge r) \rightarrow p$	$q \wedge r$	$p \rightarrow (q \wedge r)$	$((\neg q \wedge r) \rightarrow p) \leftrightarrow (p \rightarrow (q \wedge r))$
0	1	1	0	1	0	1	1	1	1
1	1	1	0	0	0	1	0	0	0
2	1	0	1	1	1	1	0	0	0
3	0	1	0	1	0	1	1	1	1
4	0	0	1	1	1	0	0	1	0
5	0	1	0	0	0	1	0	1	1
6	1	0	1	0	0	1	0	0	0
7	0	0	1	0	0	1	0	1	1

Додаток 2

Написати на будь-якій відомій студентів мові програмування програму для реалізації програмного визначення значень таблиці істинності логічних висловлювань при різних інтерпретаціях:

$$(x \wedge (y \wedge z) \Rightarrow (x \vee y \vee z))$$

Скріншоти коду для моєї програми:

```

1  #include <stdio.h>
2  #include <stdbool.h>
3  #include <stdlib.h>
4
5  int main() {
6      int x, y, z;
7      {
8          printf("Enter the value for x (0 or 1)\n");
9          scanf("%d", &x);
10         printf("Enter the value for y (0 or 1)\n");
11         scanf("%d", &y);
12         printf("Enter the value for z (0 or 1)\n");
13         scanf("%d", &z);
14         if (x != 0 & x != 1, y != 0 & y != 1, z != 0 & z != 1) {
15             printf("Sorry, wrong numbers!\n"
16                  "Please, try one more time!\n");
17             exit(0);
18         }
19         bool op0;
20         op0 = y && z;
21         printf("1. y&z = %d\n", op0);
22         bool op1;
23         op1 = x && op0;
24         printf("2. x&(y&z) = %d\n", op1);
25         bool op2;
26         op2 = x || y || z;
27         printf("3. x|y|z = %d\n", op2);
28         bool op3;
29         op3 = !op1 || op2;

```

```

30         printf("4. (x&(y&z))->(x|y|z) = %d\n", op3);
31     }
32     int rez1, rez2, rez3, rez4;
33     printf("Now I'll show you the truth table:\n"
34           "As written earlier:\n"
35           "1 = y&z\n"
36           "2 = x&(y&z)\n"
37           "3 = x|y|z\n"
38           "4 = (x&(y&z))->(x|y|z)\n"
39           "x | y | z | 1 | 2 | 3 | 4\n"
40           "--+---+---+---+---+---+---\n");
41     for(x=0; x<2; ++x)
42         for(y=0; y<2; ++y)
43             for(z=0; z<2; z++)
44                 {
45                     rez1 = y&z;
46                     rez2 = x&(y&z);
47                     rez3 = x|y|z;
48                     rez4 = !(x&(y&z)) || (x|y|z);
49                     printf("%d | %d | %d | %d | %d | %d | %d\n", x, y, z, rez1, rez2, rez3, rez4);
50                 }
51
52     return 0;
53 }

```

Результати виконання програми:

```

C:\Users\iryna\CLionProjects\dicret1\cmake-build-debug\dicret1.exe
Enter the value for x (0 or 1)
1
Enter the value for y (0 or 1)
0
Enter the value for z (0 or 1)
1
1. y&z = 0
2. x&(y&z) = 0
3. x|y|z = 1
4. (x&(y&z))->(x|y|z) = 1
Now I'll show you the truth table:
x | y | z | y&z | x&(y&z) | x|y|z | (x&(y&z))->(x|y|z) |
--+---+---+---+---+---+---+
0 | 0 | 0 | 0 | 0 | 0 | 1 |
0 | 0 | 1 | 0 | 0 | 1 | 1 |
0 | 1 | 0 | 0 | 0 | 1 | 1 |
0 | 1 | 1 | 1 | 0 | 1 | 1 |
1 | 0 | 0 | 0 | 0 | 1 | 1 |
1 | 0 | 1 | 0 | 0 | 1 | 1 |
1 | 1 | 0 | 0 | 0 | 1 | 1 |
1 | 1 | 1 | 1 | 1 | 1 | 1 |
-----
Process finished with exit code 0

```

Також в програмі присутня перевірка на коректне введення даних:

```
C:\Users\iryna\CLionProjects\dicret1\cmake-build-debug\dicret1.exe
Enter the value for x (0 or 1)
1
Enter the value for y (0 or 1)
2
Enter the value for z (0 or 1)
3
Sorry, wrong numbers!
Please, try one more time!

Process finished with exit code 0
```

Висновок

Ми на практиці ознайомились із основними поняттями математичної логіки, навчилися будувати складні висловлювання за допомогою логічних операцій та знаходити їхні істинні значення таблицями істинності, використовувати закони алгебри логіки, освоїли методи доведень.