

**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

Долінський А.Г.

Викладач:

Мельникова Н.І.

Львів – 2019р.

Тема: 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 $\neg P$, істинносне 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. У будь-якому іншому випадку його вважають істинним. Альтернативним “або” двох висловлювань P та Q називають складне висловлювання $P \oplus Q$, яке набуває істинного значення тоді і лише тоді, коли P та Q мають різні логічні значення, і є хибним в протилежному випадку. Еквівалентцією 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, і є хибним в протилежному випадку, тобто логічно еквівалентні складні висловлювання

– це висловлювання, які набувають однакових значень істинності на будь-якому наборі істиносних значень своїх складових.

Тавтологія – формула, що виконується у всіх інтерпретаціях (тотожно істинна формула). Протиріччя – формула, що не виконується у жодній інтерпретації (тотожно хибна формула). Формулу називають нейтральною, якщо вона не є ні тавтологією, ні протиріччям (для неї існує принаймні один набір пропозиційних змінних, на якому вона приймає значення Т, і принаймні один набір, на якому вона приймає значення F). Виконана формула – це формула, що не є протиріччям (інакше кажучи, вона принаймні на одному наборі пропозиційних змінних набуває значення Т).

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). Для будь-якого предиката $P(x)$ вирази $\forall x P(x)$ та $\exists x P(x)$ читаються як «всі x мають властивість $P(x)$ » та «існує (бодай один) x , що має властивість $P(x)$ » відповідно.

Додаток №1

Варіант №7

1. Формалізувати речення. Багато непорозумінь між урядами України та Польщі, але ні Україна, ні Польща не втратили економічної співпраці.
2. Побудувати таблицю істинності для висловлювань:
 $((\neg x \leftrightarrow \neg y) \leftrightarrow ((z \rightarrow (x \vee y)) \rightarrow \neg z))$.
3. Побудовою таблиць істинності вияснити, чи висловлювання є тавтологією або протиріччям: $((\neg(p \vee q)) \wedge (\neg(q \wedge r)) \rightarrow (p \vee r))$.
4. За означенням без побудови таблиць істинності та виконання еквівалентних перетворень перевірити, чи є тавтологією висловлювання: $((p \rightarrow q) \wedge (p \rightarrow \neg q)) \rightarrow (\neg p \rightarrow q)$.
5. Довести, що формули еквівалентні: $p \leftrightarrow (q \vee r)$ та $p \wedge (q \rightarrow r)$.

1. P = непорозуміння

Q = втрата економічної співпраці

x = уряд України

y = уряд Польщі

$$P(x, y) \rightarrow (\neg Q(x, y)).$$

2.

x	y	z	$\neg x$	$\neg y$	$\neg z$	$\neg x \leftrightarrow \neg y$	$x \vee y$
1	1	1	0	0	0	1	1
1	1	0	0	0	1	1	1
1	0	0	0	1	1	0	1
0	0	1	1	1	0	1	0
0	1	1	1	0	0	0	1
0	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1
0	0	0	1	1	1	1	0

$z \rightarrow (x \vee y)$	$(z \rightarrow (x \vee y)) \rightarrow \neg z$	$((\neg x \leftrightarrow \neg y) \leftrightarrow ((z \rightarrow (x \vee y)) \rightarrow \neg z))$
1	0	0
1	1	1
1	1	0
0	1	1
1	0	1
1	1	0
1	0	1
1	1	1

3.

p	q	r	$(\neg(p \vee q))$	$(\neg(q \wedge r))$
1	1	1	0	0
1	1	0	0	1
1	0	0	0	1
1	0	1	0	1
0	1	1	0	0
0	0	1	1	1
0	1	0	0	1
0	0	0	1	1

$p \vee r$	$((\neg(p \vee q)) \wedge (\neg(q \wedge r)))$	$((\neg(p \vee q)) \wedge (\neg(q \wedge r))) \rightarrow (p \vee r)$
1	0	1
1	0	1
1	0	1
1	0	1
1	0	1
1	1	1
0	0	1
0	1	0

Це висловлювання не є протиріччям чи тавтологією, воно є нейтральним.

$$4. ((p \rightarrow q) \wedge (p \rightarrow q)) \rightarrow (\neg p \rightarrow q).$$

$$((p \rightarrow q) \wedge (p \rightarrow q)) = (p \rightarrow q)$$

$$(p \rightarrow q) \rightarrow (\neg p \rightarrow q)$$

Нехай $(\neg p \rightarrow q) = F$, а $(p \rightarrow q) = T$, тільки у цьому випадку імплікація буде давати значення False.

$$(\neg p \rightarrow q) = (p \vee q)$$

$$(p \rightarrow q) = (\neg p \vee q)$$

$$(p \vee q) = F$$

$$p = F$$

$$q = F$$

$$\neg p = T$$

$$(T \vee F) \rightarrow (F \vee F) = T \rightarrow F = F.$$

Це логічне висловлювання не є тавтологією, адже може приймати значення False. Це висловлювання є нейтральним.

5.

$p \leftrightarrow (q \vee r)$ та $p \wedge (q \rightarrow r)$.

$p \leftrightarrow (q \vee r)$

p	q	r	$q \vee r$	$p \leftrightarrow (q \vee r)$
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	0	0
0	1	1	1	0
0	0	1	1	0
0	1	0	1	0
0	0	0	0	0

$p \wedge (q \rightarrow r)$

p	q	r	$q \rightarrow r$	$p \wedge (q \rightarrow r)$
1	1	1	1	1
1	1	0	0	0
1	0	1	1	1
1	0	0	1	1
0	1	1	1	0
0	0	1	1	0
0	1	0	0	0
0	0	0	1	0

Ці дві формули не є еквівалентними.

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

$$((\neg x \leftrightarrow \neg y) \leftrightarrow ((z \rightarrow (x \vee y)) \rightarrow \neg z)).$$

```
int x, y, z;
int r1, r2, r3, r4, r5, r6, r7, r8;
setlocale(LC_ALL, "Ukrainian");
printf("(( -x <-> -y) <-> (( z -> (x v y)) -> -z))\n");
printf("Введіть значення x (1 або 0):\n ");
scanf_s("%d", &x);
printf("Введіть значення y (1 або 0):\n ");
scanf_s("%d", &y);
printf("Введіть значення z (1 або 0):\n ");
scanf_s("%d", &z);

if((x == 0 || x == 1) && (y == 0 || y == 1) && (z == 0 || z == 1)){
    r1 = !x;
    r2 = !y;
    r3 = !z;
    r4 = r1 == r2;
    r5 = x || y;
    r6 = r3 || r5;
    r7 = !r6 || r3;
    r8 = r4 == r7;
    printf("-x = %d\n", r1);
    printf("-y = %d\n", r2);
    printf("-z = %d\n", r3);
    printf("(-x <-> -y) = %d\n", r4);
    printf("(x v y) = %d\n", r5);
    printf("( z -> (x v y)) = %d\n", r6);
    printf("(( z -> (x v y)) -> -z) = %d\n", r7);
    printf("(( -x <-> -y) <-> (( z -> (x v y)) -> -z)) = %d\n", r8);
}
else{
    printf("Помилка вводу!");
}
return 0;
}
```

Скрін-шот усього коду реалізації програми на мові C.

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
```

Спочатку підключаємо потрібні нам бібліотеки.

```
int main() {
    int x, y, z;
    int r1, r2, r3, r4, r5, r6, r7, r8;
```

Створюємо головну функцію main.

Визначаємо змінні, що будуть зберігати інформацію типу integer.

```
setlocale(LC_ALL, "Ukrainian");
printf("(( -x <-> -y) <-> (( z -> (x v y)) -> -z))\n");
printf("Введіть значення x (1 або 0):\n ");
scanf_s("%d", &x);
printf("Введіть значення y (1 або 0):\n ");
scanf_s("%d", &y);
printf("Введіть значення z (1 або 0):\n ");
scanf_s("%d", &z);
```

Підключаємо українську мову. Скануємо дані вводу від користувача програми.

```
if((x == 0 || x == 1) && (y == 0 || y == 1) && (z == 0 || z == 1)){
    r1 = !x;
    r2 = !y;
    r3 = !z;
    r4 = r1 == r2;
    r5 = x || y;
    r6 = r3 || r5;
    r7 = !r6 || r3;
    r8 = r4 == r7;
    printf("-x = %d\n", r1);
    printf("-y = %d\n", r2);
    printf("-z = %d\n", r3);
    printf("(-x <-> -y) = %d\n", r4);
    printf("(x v y) = %d\n", r5);
    printf("( z -> (x v y)) = %d\n", r6);
    printf("(( z -> (x v y)) -> -z) = %d\n", r7);
    printf("(( -x <-> -y) <-> (( z -> (x v y)) -> -z)) = %d\n", r8);
}
else{
    printf("Помилка вводу!");
}
return 0;
}
```

За допомогою умовного оператора if вводимо умову вводу. Якщо користувач вводить коректні дані, то реалізовуємо частину програми, що міститься в if. Змінні r1-r8 зберігають результати майбутньої таблиці, яку ми по-частинно виводимо за допомогою printf. При хибному вводі, виводимо на екран “Помилка вводу!”. Укінці не забуваємо завершити виконання функції main за допомогою return, що повертає нічого.

Проведемо декілька тестів.

```
Microsoft Visual Studio Debug Console

(( -x <-> -y) <-> (( z -> (x v y)) -> -z))
Введ?ть значення x (1 або 0):
1
Введ?ть значення y (1 або 0):
0
Введ?ть значення z (1 або 0):
1
-x = 0
-y = 1
-z = 0
(-x <-> -y) = 0
(x v y) = 1
( z -> (x v y)) = 1
(( z -> (x v y)) -> -z) = 0
(( -x <-> -y) <-> (( z -> (x v y)) -> -z)) = 1
```

Вводимо значення $x = 1$, $y = 0$, $z = 1$.

Результат показаний на скрін-шоті.

Microsoft Visual Studio Debug Console

```
(( -x <-> -y) <-> (( z -> (x v y)) -> -z))  
Введіть значення x (1 або 0):  
0  
Введіть значення y (1 або 0):  
0  
Введіть значення z (1 або 0):  
0  
-x = 1  
-y = 1  
-z = 1  
(-x <-> -y) = 1  
(x v y) = 0  
( z -> (x v y)) = 1  
(( z -> (x v y)) -> -z) = 1  
(( -x <-> -y) <-> (( z -> (x v y)) -> -z)) = 1
```

Проведемо ще один тест із іншими вхідними даними ($x = 0, y = 0, z = 0$).

І також проведемо тест із вводом хибних даних.

Microsoft Visual Studio Debug Console

```
(( -x <-> -y) <-> (( z -> (x v y)) -> -z))  
Введіть значення x (1 або 0):  
1  
Введіть значення y (1 або 0):  
0  
Введіть значення z (1 або 0):  
2  
Помилка вводу!
```

При $z = 2$, програма видає помилку вводу.

Microsoft Visual Studio Debug Console

```
(( -x <-> -y) <-> (( z -> (x v y)) -> -z))  
Введіть значення x (1 або 0):  
1  
Введіть значення y (1 або 0):  
р  
Введіть значення z (1 або 0):  
Помилка вводу!
```

При введенні типу `char`, а не `integer` програма також видає помилку вводу.

Висновок: Я ознайомився на практиці із основними поняттями математичної логіки, навчився будувати складні висловлювання за допомогою логічних операцій та знаходити їхні істинні значення

таблицями істинності, використовував закони алгебри логіки, освоїв методи доведень.