МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту

Лабораторна робота № 1

з дисципліни

«Дискретна математика»

Виконала:

студентка групи КН-114

Кемська Юлія

Викладач:

Мельникова Н.І.

Львів – 2019

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 істинноносні значення таблицями істинності, використовувати закони алгебри логіки, освоїти методи доведень.

1.1. Основні поняття математичної логіки. Логічні операції

Просте висловлювання (атомарна формула, атом) – це розповідне речення, про яке можна сказати, що воно істинне (T 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 $\Leftrightarrow$).

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 T, і принаймні один набір, на якому вона приймає значення F).

1.2. Закони логіки висловлювань

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: $P \vee (\overline{P}) = \mathrm{T}$	закон протиріччя: $P \wedge (\overline{P}) = \mathrm{F}$
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)	
$P \vee \mathrm{T} = \mathrm{T}$	$P \wedge \mathrm{T} = P$(тот)
$P \vee \mathrm{F} = P$ (тот)	$P \wedge \mathrm{F} = \mathrm{F}$

1.3. Логіка першого ступеня. Предикати і квантори. Закони логіки першого ступеня

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 x1,..., xn.

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.

Основні закони логіки першого ступеня (логіки предикатів):

1. $\neg(\forall x P(x)) = \exists x(\neg P(x))$, $\forall x P(x) = \neg\exists x(\neg P(x))$.
2. $\neg(\exists x P(x)) = \forall x(\neg P(x))$, $\exists x P(x)) = \neg\forall x(\neg P(x))$.
3. $\forall x(P(x) \wedge Q(x)) = \forall x P(x) \wedge \forall x Q(x)$.
4. $\exists x(P(x) \vee Q(x)) = \exists x P(x) \vee \exists x Q(x)$
5. $\forall x(P(x) \wedge Q) = \forall x P(x) \wedge Q$.
6. $\forall x(P(x) \vee Q) = \forall x P(x) \vee Q$
7. $\exists x (P (x) \wedge Q) = \exists x P (x) \wedge Q$.
8. $\exists x (P (x) \vee Q) = \exists x P (x) \vee Q$.
9. $\forall x \forall y P (x, y) = \forall y \forall x P (x, y)$.
10. $\exists x \exists y P(x, y) = \exists y \exists x P(x, y)$.
11. $\forall x P(x) = \forall t P(t)$, $\exists x P(x) = \exists t P(t)$.
12. $\forall x P = P$, $\exists x P = P$.

1.4. Методи доведень

Метод «від протилежного». У допущенні, що висловлювання P істинне, а Q хибне, використовуючи аргументоване міркування, одержимо протиріччя. Цей спосіб заснований на тому, що імплікація (P $\Rightarrow$Q) набуває хибного значення лише тоді, коли P істинне, а Q хибне.

Варіант № 9

Додаток 1.

1. Формалізувати речення.

Іван прийде на іспит то отримає оцінку відмінно, якщо Іван не прийде на іспит тоді він і Сергій отримає позитивну оцінку.

P- прийти на іспит

Q – отримати оцінку

x- Іван, y- Сергій, r- оцінка відмінно, z- оцінка позитивна

$P(x) \to Q(x,r)$

$\neg P(x) \to Q(x,y,z)$

2. Побудувати таблицю істинності для висловлювань:
$(x \to (y \wedge z)) \to (x \to (y \wedge z))$;

x	y	z	y^z	x→(y^z)	y^z	x→(y^z)	(x→(y^z))→(x→(y^z))
T	T	T	T	T	T	T	T
T	T	F	F	F	F	F	T
T	F	T	F	F	F	F	T
T	F	F	F	F	F	F	T
F	T	T	T	T	T	T	T
F	T	F	F	T	F	T	T
F	F	T	F	T	F	T	T
F	F	F	F	T	F	T	T

3. Побудовою таблиць істинності вияснити, чи висловлювання є тавтологією або протиріччям: $((p \wedge q) \wedge (\neg q \to r)) \leftrightarrow (\neg p \vee \neg r)$.

p	q	r	┐p	┐q	┐r	p^q	┐q→r	┐p∨┐r	(p^q)^(┐q→r)	((p^q)^(┐q→r))↔(┐p∨┐r)
T	T	T	F	F	F	T	T	F	T	F
T	T	F	F	F	T	T	T	T	T	T
T	F	T	F	T	F	F	T	F	F	T
T	F	F	F	T	T	F	F	T	F	F
F	T	T	T	F	F	F	T	T	F	F
F	T	F	T	F	T	F	T	T	F	F
F	F	T	T	T	F	F	T	T	F	F
F	F	F	T	T	T	F	F	T	F	F

Відповідь: формула не є ні тавтологією, ні протиріччям , отже вона є **нейтральною .**

4. За означенням без побудови таблиць істинності та виконання еквівалентних перетворень перевірити, чи є тавтологією висловлювання: $((p\to q)\wedge(\neg p\to q))\to q$.

$q=F$; $(p\to q)\wedge(\neg p\to q)=T$

$(p\to q)=T$; $(\neg p\to q)=T$ (бо між ними стоїть знак кон'юнкції)

$(F\to F)=T$; $(T\to F)=F$.

$T\wedge F=F$

Відповідь: висловлення є тавтологією.

5. Довести,щоформулиеквівалентні:
$(p\vee\neg p)\to(\neg p\vee q)$ та $(r\to q)\wedge(p\to r)$.

p	q	r	$\neg p$	$p\vee\neg p$	$\neg p\vee q$	$(p\vee\neg p)\to(\neg p\vee q)$	$r\to q$	$p\to r$	$(r\to q)\wedge(p\to r)$.
T	T	T	F	T	T	T	T	T	T
T	T	F	F	T	T	T	T	F	F
T	F	T	F	T	F	F	F	T	F
T	F	F	F	T	F	F	T	F	F
F	T	T	T	T	T	T	T	T	T
F	T	F	T	T	T	T	T	T	T
F	F	T	T	T	T	T	F	T	F
F	F	F	T	T	T	T	T	T	T

Відповідь: по 7-мому і 10-тому стовпцях ми бачимо, що формули *не є еквівалентними.*

Додаток 2.

$(x\to(y\wedge z))\to(x\to(y\wedge z))$

Код програми:

```cpp
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <iostream>

using namespace std;
//функція перевірки конкретності введених даних
int ok(int a){
    if(a==1 || a==0)
    return 1; return 0;
}
//функція імплікації
int impl(int p, int q){
    if (p==1 && q==0)
    return 0;
    return 1;
}
int main ()
{
    int x=-1, y=-1, z=-1;
    //Введення даних
    while(ok(y)*ok(x)*ok(z)==0){
      cout<<"Enter x, y and z like 0 or 1"<<endl;
      cin>>x>>y>>z;
      if (ok(y)*ok(x)*ok(z)==0)
      cout<<"Enter correct numbers"<<endl;
    }
    //Вивід таблиці істиності для введених даних
      cout<<"| x | y | z | (y/\\z) | x=>(y/\\z) | (x=>(y/\\z))=>(x=>(y/\\z)) |"<<endl;

       cout<<"| "<<x<<" | "<<y<<" | "<<z<<" |   "<<(y&z)<<"    |    "<<impl(x,y&z)<<"     |       "<< impl(impl(x,y&z),impl(x,y&z))<<"     |";
}
```

Результат виконання програми

```
C:\Users\Admin\source\repos\labaulia\bin\Debug\labaulia.exe
Enter x,y and z like 0 or 1
5 4 5
Enter correct numbers
Enter x,y and z like 0 or 1
0 0 1
| x | y | z | (y/\z) | x=>(y/\z) | (x=>(y/\z))=>(x=>(y/\z))|
| 0 | 0 | 1 |    0    |     1      |              1              |
Process returned 0 (0x0)   execution time : 11.520 s
Press any key to continue.
```

Висновок: Я ознайомилась на практиці із основними поняттями математичної логіки, навчилась будувати складні висловлювання за допомогою логічних операцій та знаходити їхні істинноносні значення таблицями істинності, використовувати закони алгебри логіки, освоїла методи доведень, використовуючи теоретичні знання, здобуті на лекції. Моя програма автоматично знаходить істиноносні значення складного висловлювання для всіх інтерпретацій простих висловлювань, дозволяє ввести данні вручну (кількість простих висловлювань, логічні операції), здійснює перевірку на некоректне введення даних.