МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Кіровоградський державний педагогічний університет
Імені Володимира Винниченка

З.П. Халецька, В.В. Нарадовий

МАТЕМАТИЧНА ЛОГІКА ТА ТЕОРІЯ АЛГОРИТМІВ

Кропивницький – 2016

ББК 22.12:73
УДК 510.5

З.П. Халецька, В.В. Нарадовий
Математична логіка та теорія алгоритмів: Навчальний посібник. – Кропивницький: РВВ КДПУ ім.. В. Винниченка, 2017. – 128 с.
ISBN

Даний посібник охоплює матеріал модулів «Алгебра та числення висловлень», «Логіка предикатів», «Теорія алгоритмів» відповідно до програми курсу «Математична логіка та теорія алгоритмів». Викладення теоретичного матеріалу ілюструється прикладами. В кожному розділі після теоретичного блоку наведено запитання для самоконтролю та вправи для самостійного розв'язання.
Посібник рекомендується для студентів всіх спеціальностей денної та заочної форми навчання, які вивчають математичну логіку та теорію алгоритмів. Він може бути корисним вчителям математики та інформатики.

Рецензенти: Безущак О.О., заслужений працівник освіти, кандидат фізико-математичних наук, доцент кафедри алгебри та математичної логіки, Київський національний університет імені Тараса Шевченка;
Якименко С.М., кандидат фізико-математичних наук, завідувач кафедри вищої математики та фізики, Кіровоградський національний технічний університет.

Затверджено до друку Вченою Радою Кіровоградського державного педагогічного університету імені Володимира Винниченка (протокол № 7 від "30" січня 2017 року)

ISBN

ЗМІСТ

ПЕРЕДМОВА 5
1. АЛГЕБРА ВИСЛОВЛЕНЬ 6
1.1. ВИСЛОВЛЕННЯ ТА ОПЕРАЦІЇ НАД НИМИ. ФОРМУЛИ АЛГЕБРИ ВИСЛОВЛЕНЬ 6
1.2 ВІДНОШЕННЯ РІВНОСИЛЬНОСТІ НА БАЗІ АЛГЕБРИ ВИСЛОВЛЕНЬ 14
1.3 НОРМАЛЬНІ ФОРМИ ФОРМУЛ АЛГЕБРИ ВИСЛОВЛЕНЬ ТА ЇХ ЗАСТОСУВАННЯ. 18
1.4 БУЛЕВІ ФУНКЦІЇ. ФУНКЦІОНАЛЬНА ПОВНОТА. ПОЛІНОМИ ЖЕГАЛКІНА 23
1.5 ТЕХНІЧНІ ЗАСТОСУВАННЯ АЛГЕБРИ БУЛЕВИХ ФУНКЦІЙ 31
1.6 ВІДНОШЕННЯ ЛОГІЧНОГО НАСЛІДКУ 37
2. ЧИСЛЕННЯ ВИСЛОВЛЕНЬ 49
2.1 АЛФАВІТ, ФОРМУЛИ, АКСІОМИ, ПРАВИЛА ВИВОДУ ЧИСЛЕННЯ ВИСЛОВЛЕНЬ L_1. ПРИКЛАДИ ДОВЕДЕНЬ В ЧИСЛЕННІ L_1 50
2.2. ВИВІДНІСТЬ ІЗ ГІПОТЕЗ. МЕТАТЕОРЕМА ДЕДУКЦІЇ. ДОДАТКОВІ ПРАВИЛА ВИВОДУ 53
3. ЛОГІКА ПРЕДИКАТІВ 59
3.1. ПРЕДИКАТИ ТА ОПЕРАЦІЇ НАД НИМИ. 59
3.2. ФОРМУЛИ ЛОГІКИ ПРЕДИКАТІВ. ІНТЕРПРЕТАЦІЯ ФОРМУЛ. ВИКОНУВАНІ ТА ЛОГІЧНО ЗАГАЛЬНОЗНАЧУЩІ ФОРМУЛИ 67
3.3. ВІДНОШЕННЯ РІВНОСИЛЬНОСТІ В ЛОГІЦІ ПРЕДИКАТІВ. ЗВЕДЕНА ТА ПРЕНЕКСНА НОРМАЛЬНІ ФОРМИ 73
3.4. МЕТОД РЕЗОЛЮЦІЙ 79
3.5. ЗАСТОСУВАННЯ ЛОГІКИ ПРЕДИКАТІВ 87
3.5.1 Застосування символіки логіки предикатів в математичних формулюваннях 87
3.5.2 Використання елементів математичної логіки для аналізу структури математичних теорем 88

4. ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ..94
4.1. ПОНЯТТЯ АЛГОРИТМУ. 94
4.2. НОРМАЛЬНІ АЛГОРИТМИ 94
4.3. РЕКУРСИВНІ ФУНКЦІЇ...95
4.4. МАШИНА ТЬЮРИНГА 104
4.5. МАШИНИ З НАТУРАЛЬНОЗНАЧНИМИ РЕГІСТРАМИ 121
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 126
ПРЕДМЕТНИЙ ПОКАЗЧИК 127

ПЕРЕДМОВА

Даний посібник є переробленим та доповненим виданням посібника «Математична логіка та теорія алгоритмів», що був виданий у 2009 році колективом авторів В.М. Євладенко, З.П. Халецька, В.В. Нарадовий, та протягом семи років використовувався для студентів спеціальностей «Математика*», «Інформатика» та «Статистика» фізико-математичного факультету Кіровоградського державного педагогічного університету імені Володимира Винниченка.

Даний посібник охоплює матеріал модулів «Алгебра та числення висловлень», «Логіка предикатів», «Теорія алгоритмів» відповідно до програми курсу «Математична логіка та теорія алгоритмів». Викладення теоретичного матеріалу ілюструється прикладами. В кожному розділі після теоретичного блоку наведено запитання для самоконтролю та вправи для самостійного розв’язання.

До кожної теми у посібнику подано формулювання означень основних понять, теорем та інших фактів, знання яких необхідне для опанування цієї теми. Наводяться розв’язані приклади з детальним поясненням, що дозволить читачам самостійно опрацьовувати даний матеріал і краще розуміти деякі, традиційно складні для студентів, теми.

До кожної теми розділів підібрані вправи для самостійного розв’язування. Їх кількість достатня для роботи на практичних заняттях та для виконання поточних домашніх завдань. Крім того, кожна тема завершується запитаннями для самоконтролю, що допоможе студентам при підготовці до модульних та підсумкового контролів.

Для зручності користування в кінці посібника вміщено предметний покажчик.

Посібник рекомендується для студентів денної та заочної форм навчання спеціальностей «Математика*», «Статистика», а також він буде корисним студентам спеціальності «Інформатика» при вивченні даних розділів у курсах «Дискретна математика» та «Теорія алгоритмів та математична логіка».

1. АЛГЕБРА ВИСЛОВЛЕНЬ

1.1. ВИСЛОВЛЕННЯ ТА ОПЕРАЦІЇ НАД НИМИ. ФОРМУЛИ АЛГЕБРИ ВИСЛОВЛЕНЬ

Висловлення – вихідне, неозначувальне поняття алгебри висловлень. Під *висловленням* розуміють таке речення, про яке можна говорити, що воно істинне чи хибне.

У логіці висловлень висловлення розглядаються лише з точки зору їх істинності чи хибності; ніякими іншими властивостями висловлень не цікавляться; значення істинності чи хибності висловлень не аналізується, а береться як дані. Відповідь про істинність чи хибність висловлення дає галузь науки чи людської діяльності, до якої воно належить.

У подальшому дотримуватимемось точки зору двозначної класичної логіки, в якій приймаються два основних припущення:

1) кожне висловлення є або істинним, або хибним, тобто третього не дано (закон виключеного третього);

2) жодне висловлення не є одночасно істинним і хибним (закон виключення суперечності або закон протиріччя).

Зазначимо, що сучасна логіка розглядає так звані багатозначні логіки, де ці закони місця не мають.

Висловлення бувають простими і складеними. Прості висловлення це висловлення, які не можна подати як логічну композицію більш коротких висловлень (їм відповідають прості розповідні речення). Їх позначають малими літерами латинського алфавіту *a,b,c,…* або тими ж буквами з індексами внизу, та називають *пропозиційними* (логічними, висловлювальними) буквами або змінними.

Наприклад: a: «Київ – столиця України»; b: «Пряма $x+2y-6=0$ проходить через точку з координатами (2,2)»; c: «Аристотель – український філософ»; d: «Простих чисел нескінченна кількість»; *p*: «Річка Ніл впадає в Чорне море».

Домовимось записувати $|a|=1$, якщо висловлення *a* – істинне і $|a|=0$, якщо висловлення *a* – хибне. У першому випадку говорять, що значення істинності висловлення *a* (значення функції істинності для даного значення аргументу) дорівнює 1, а в другому, що значення істинності висловлення *a* дорівнює 0.

Для наведених прикладів висловлень маємо наступні логічні значення:

$|a| = 1$, $|b| = 1$, $|c| = 0$, $|d| = 1$, $|p| = 0$.

Складені (складні) висловлення утворюються з простих за допомогою логічних зв'язок (операцій). Розглядатимемо тільки істинно-функціональні комбінації висловлень, в яких істинність чи хибність нових висловлень однозначно визначається істинністю чи хибністю складових.

Основними логічними операціями є *заперечення, кон'юнкція*, *диз'юнкція, імплікація* та *еквіваленція.*

Запереченням висловлення a називається нове висловлення $\neg a$ або $\overline{a}$ (читається "не a"), яке істинне, якщо вихідне висловлення a – хибне і хибне, якщо $|a| =1$. Зв'язок логічних значень висловлень $\overline{a}$ та a ілюструємо в наступній таблиці істинності:

Табл.1. Таблиця істинності операції заперечення.

$\|a\|$	$\|\overline{a}\|$
0	1
1	0

Пропозиційні змінні та їх заперечення називають *літералами*.

Кон'юнкцією двох висловлень a і b, називається нове висловлення $a \wedge b$, яке істинне лише в тому випадку, коли обидва висловлення a і b є істинними, і хибне, коли хоч одне з висловлень a або b хибне. Еквівалентні назви – «логічний добуток», «і», «AND».

Диз'юнкцією двох висловлень a і b називається нове висловлення $a \vee b$, яке істинне в тих випадках, коли хоч одне з висловлень a або b є істинним, і хибне тільки в тому випадку, коли обидва висловлення a і b є хибними. Еквівалентні назви – «логічна сума», «або», «OR».

Імплікацією двох висловлень a і b називається нове висловлення $a \rightarrow b$, яке є хибним в одному випадку, а саме, коли висловлення a істинне, а b – хибне, а в інших випадках $/a \rightarrow b/=1$. У висловленні $a \rightarrow b$ висловлення a називається умовою або *антецедентом*, а висловлення b – наслідком або *консеквентом*. Умова b є необхідною для виконання a, а умова a є достатньою для виконання b.

Еквіваленцією двох висловлень a і b, називається нове висловлення $a \leftrightarrow b$, яке істинне лише в тому випадку, коли обидва висловлення a і b набувають однакових значень істинності – вони або обидва істинні, або обидва хибні, в інших випадках $/a \leftrightarrow b/=0$. Умова b є необхідною і достатньою для виконання a.

Табл.2 Таблиця істинності бінарних логічних операцій.

$\lvert a\rvert$	$\lvert b\rvert$	$\lvert a\wedge b\rvert$	$\lvert a\vee b\rvert$	$\lvert a\to b\rvert$	$\lvert a\leftrightarrow b\rvert$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Кожне із введених означень операцій над висловленнями можна розглядати як означення деякої дії над символами 0 і 1, тобто як визначення деякої операції (функції) на двохелементній множині $\{0,1\}$. Це дає можливість записати рівності для обрахунку логічних значень висловлень: $\overline{a}$, $a\wedge b$, $a\vee b$, $a\to b, a\leftrightarrow b$, відповідно. $\lvert\overline{a}\rvert=\overline{\lvert a\rvert}$, $\lvert a*b\rvert=\lvert a\rvert*\lvert b\rvert$, де знак * означає один із символів операцій $\wedge,\vee,\to,\leftrightarrow$.

Пропозиційні букви, символи логічних операцій та дужки становлять вихідні символи мови алгебри висловлень, тобто її *алфавіт*. Будь-яка скінченна послідовність вихідних символів є *виразом* або *словом* мови. З множини всіх слів виділяють так звані формули, які в мові алгебри висловлень відповідають реченням природної мови.

Формулами алгебри висловлень називаються пропозиційні букви і вирази виду $(\overline{A}),(A\wedge B),(A\vee B),(A\to B),(A\leftrightarrow B)$, де A, B - формули. При цьому пропозиційні букви називають *елементарними* або *атомарними формулами*.

Очевидно, що за скінченне число кроків завжди можна встановити, чи є той чи інший вираз формулою логіки висловлень. Означення формули вимагає, щоб кожній логічній операції відповідала пара дужок. Дужки у формулі однозначно визначають порядок виконання операцій. Проте наявність великої кількості дужок робить запис формули досить громіздким. Для спрощення запису формул користуються такими правилами:

1) зовнішні дужки в записі формули опускають;
2) логічним операціям приписують певний ранг (старшинство операцій), згідно якого операції виконуються в такому порядку: $\neg, \wedge, \vee, \to, \leftrightarrow$.

Приклад 1. Обчислити значення істинності формули A=$a\to\overline{b}\leftrightarrow a\wedge\overline{c}$ якщо $|a| = 1$, $|b| = 1$, $|c| = 1$.

<u>Розв'язання</u>. Дана формула є скороченим записом наступної формули: $((a\to\overline{b})\leftrightarrow(a\wedge(\overline{c})))$. Послідовно знаходимо $|\overline{b}|$=0, $|a\to\overline{b}|=1\to 0$=0, $|\overline{c}|$=0, $|a\wedge\overline{c}|=\ 1\wedge 0=0$, $|a\to\overline{b}\leftrightarrow a\wedge\overline{c}|=0\leftrightarrow 0$=1.

Отже, при даній інтерпретації логічних змінних значення істинності формули дорівнює 1.

Результати обчислень значень істинності формули для всіх можливих наборів значень істинності пропозиційних змінних зручно подавати у вигляді таблиці, яку називають таблицею істинності формули.

Так, для формули A= $a \to \bar{b} \leftrightarrow a \wedge \bar{c}$ таблиця істинності матиме такий вигляд:

$\|a\|$	$\|b\|$	$\|c\|$	$\|\bar{b}\|$	$\|a \to \bar{b}\|$	$\|\bar{c}\|$	$\|a \wedge \bar{c}\|$	$\|a \to \bar{b} \leftrightarrow a \wedge \bar{c}\|$
0	0	0	1	1	1	0	0
0	0	1	1	1	0	0	0
0	1	0	0	1	1	0	0
0	1	1	0	1	0	0	0
1	0	0	1	1	1	1	1
1	0	1	1	1	0	0	0
1	1	0	0	0	1	1	0
1	1	1	0	0	0	0	1

Кожному символу логічної операції в формулі **A** виділено окремий стовпчик таблиці істинності.

Зазначимо, що кожний стовпчик таблиці для формули A відповідає певному кроку процесу побудови формули A (певній підформулі). Фактично ж таблицю істинності формули утворюють стовпчики значень істинності пропозиційних змінних (перших три стовпчики) і останній стовпчик, який визначає функцію істинності даної формули. Інші стовпчики (стовпчики істинності підформул) носять допоміжний характер.

Таблиця істинності формули $A(a_1, a_2, ... a_n)$ містить 2^n рядків (кожна пропозиційна змінна незалежно від інших може набувати одного з двох значень: 0 або 1, тобто число всіх можливих різних наборів значень *n* пропозиційних змінних є 2^n).

Формули алгебри висловлень поділяються на наступні типи: тавтології (тотожно істинні), суперечності (тотожно хибні) та нейтральні.

Формула алгебри висловлень $A(a_1, a_2, ... a_n)$ називається *тавтологією*, якщо при всіх можливих значеннях пропозиційних букв $a_1, a_2, ..., a_n$ вона істинна, тобто її функція істинності тотожно дорівнює 1. Позначення: $\models A(a_1, a_2, ... a_n)$.

Формула логіки висловлень, значення істинності якої дорівнює 0 при всіх можливих значеннях істинності її пропозиційних змінних називається *суперечністю*. Отже, суперечності – це такі формули, значення істинності яких не дорівнює 1 при жодній інтерпретації.

Формула, яка не є ні тавтологією, ні суперечністю, називається *нейтральною*.

Будь-яку формулу, яка не є суперечністю називають *виконуваною*, а не тотожно істинну – *спростовною*.

Якщо формула А залежить від невеликої кількості змінних, то для з'ясування її типу зручно користуватися таблицями істинності. У деяких випадках доцільно застосовувати метод міркувань від супротивного.

Приклад 2. Довести, що якщо формули $\overline{A} \to \overline{B}$ і $\overline{A} \to B$ тавтології, то формула А – також тавтологія.

Розв'язання. Оскільки А і В –довільні формули, то ми не можемо використати таблиці істинності. Застосуємо метод від супротивного. Припустимо, що А – не тавтологія, тобто існує набір значень пропозиційних змінних, від яких залежить А, на якому А набуває логічного значення 0. Тобто $|A| = 0$. Тоді $|\overline{A}| = 1$. За умовою $|\overline{A} \to \overline{B}| = 1$ і $|\overline{A} \to B| = 1$. Звідки $|\overline{B}| = 1$ і одночасно $|B| = 1$, що неможливо. Отже, припущення невірне.

Запитання для самоконтролю

1. Що називать висловленням?
2. Які припущення лежать в основі класичної двозначної логіки?
3. Що називають пропозиційними буквами (змінними)?
4. Як називаються основні логічні операції алгебри висловлень? Дайте означення кожної з них.
5. Що називають літералом в алгебрі висловлень?
6. Які символи утворюють алфавіт алгебри висловлень?
7. Що називають формулою алгебри висловлень?
8. Які правила щодо скорочення дужок застосовуються у формулах алгебри висловлень?
9. Як будується таблиця істинності формули $A(a_1, a_2, ...a_n)$? Скільки рядків вона містить?

10. Яка формула називається а) тавтологією, б) суперечністю, в) виконуваною, г) спростовною, д) нейтральною? Наведіть приклади формул кожного типу.
11. Якщо формула $A(a_1, a_2, ... a_n)$ тавтологія, якою буде формула $\overline{A(a_1, a_2, ... a_n)}$?
12. Якщо формула $A(a_1, a_2, ... a_n)$ нейтральна, якою буде формула $\overline{A(a_1, a_2, ... a_n)}$?
13. Якщо формула $A(a_1, a_2, ... a_n)$ суперечність, якою буде формула $\overline{A(a_1, a_2, ... a_n)}$?
14. Скільки різних наборів пропозиційних змінних існує для формули $A(a_1, a_2, ... a_n)$?
15. Якщо формули A, $A \to B$ тавтології, то що можна сказати про формулу B?

<u>ВПРАВИ.</u>

1.1 Які з наступних речень є висловленнями?

a) до університету вступають найкращі учні;

b) для будь-якого дійсного числа x: $|sin\ x| \le 1$;

c) математична логіка;

d) подивіться в підручник;

e) 3 – парне число;

f) Київ – столиця України;

g) взимку завжди йде сніг.

1.2 Записати висловлення: «Числа 10 і 8 є парними числами» за допомогою таких простих висловлень: “число 10 – парне” – a, “число 8 – непарне” –b.

1.3 Прочитайте наступні формули природною мовою: а) $c \to \overline{d}$, б) $a \wedge c \to b$, в) $b \to \overline{a}$, г) $a \wedge c \to d$, д) $b \vee c \vee d \leftrightarrow a$, якщо *a:* «цей чотирикутник – паралелограм», *b:* «цей чотирикутник – ромб», *c:* «цей чотирикутник – прямокутник», *d:* «цей чотирикутник – квадрат».

1.4 Записати висловлення у вигляді формули, позначивши літерами атомарні висловлення, тобто не побудовані з інших висловлень:

a) якщо собака гавкає, то господар не спить, і якщо господар не спить, то собака гавкає;

b) або Петро піде на заняття, і Дмитро туди не піде, або Петро не піде на заняття, і Дмитро відмінно проведе час;

c) необхідна й достатня умова для щастя студента полягає в тому, щоб пити пиво, багато спати і насолоджуватися музикою;

d) Іван дивиться телевізор тільки в тому випадку, коли там показують бойовик;

e) корінь із цілого числа є або цілим, або ірраціональним числом;

f) для того, щоб X було непарним, достатньо, щоб X було простим;

g) необхідною умовою того, що послідовність збігається, є її обмеженість.

1.5 Записати висловлення у вигляді формули і побудувати таблицю істинності:

a) якщо число ділиться на 5 і не ділиться на 2, то воно не ділиться на 10;

b) Петро вступить до університету тоді і тільки тоді, коли буде добре вчитися і отримає високий бал на вступних іспитах;

c) якщо після дощу ясно, то на небі з'явиться веселка і буде тепло;

d) якщо дитина плаче, то вона або хоче їсти, або в неї щось болить;

e) студент виконує завдання довго тоді і тільки тоді, коли вони складні або він погано знає матеріал;

f) якщо число просте, то воно ділиться або на 1, або саме на себе;

g) якщо протягом семестру я буду гарно вчитися і матиму хороші оцінки, то здам успішно сесію і буду отримувати стипендію.

1.6 За допомогою таблиці істинності визначити тип формули**:**

a) $a \to b \wedge \overline{a}$;

b) $a \vee b \leftrightarrow \overline{b}$;

c) $(b \to (a \to c)) \wedge (b \to a) \wedge b \wedge \overline{c}$;

d) $\overline{a} \vee b \leftrightarrow a \wedge \overline{c} \to b$;

e) $(a \to (b \to c)) \leftrightarrow (a \wedge b \to c)$;

f) $(a \to b) \vee (\overline{b} \leftrightarrow c) \leftrightarrow b \wedge \overline{c}$;

g) $(a \leftrightarrow b) \to ((a \to a \wedge c) \leftrightarrow (b \to b \wedge c))$;

h) $(a \to b) \wedge (c \leftrightarrow \overline{d}) \wedge (b \leftrightarrow \overline{c}) \wedge (\overline{a} \to d)$.

1.7 Довести, що якщо формули $A \vee B$ і $\overline{A} \vee C$ є тавтологіями, то $B \vee C$ також тавтологія.

1.8 За допомогою методу від супротивного переконатися, чи будуть тавтологіями наступні формули:

a) $(a \to b) \to (a \vee c \to b \vee c)$;

b) $((\overline{a} \vee b) \wedge (a \vee \overline{b})) \to ((a \to b \vee c) \to (a \to c))$;

c) $((p \wedge q) \to r) \wedge ((p \vee q) \to \overline{r}) \to (p \wedge q \wedge r)$;

d) $((p \to q) \wedge (r \to s)) \to ((p \wedge r) \to (q \wedge s))$.

1.9 Розглядається справа Брауна, Джонса і Сміта. Один з них здійснив злочин. На слідстві кожен з них зробив дві заяви.

Браун: «Я не винен. Сміт зробив це.»

Джонс: «Сміт не винен. Браун зробив це.»

Сміт: «Я не робив цього. Джонс не робив цього.»

Суд встановив, що один з них двічі сказав неправду, другий двічі сказав правду, третій – один раз правду, другий раз збрехав. Хто здійснив злочин?

1.10 Чотири спортсмени – Андрій, Іван, Роман і Сергій в змаганнях посіли перші чотири місця, причому жодне з місць не було поділене між ними. Про зайняті місця є інформація: «Друге місце посів Роман, а третє Сергій», «Переміг Роман, Іван був другим», «Друге місце посів Андрій, а Сергій останнє». Як розподілились місця між учасниками змагань, якщо відомо, що в кожному з трьох висловлень про зайняті місця одне твердження істинне, а друге хибне?

1.11 При яких натуральних *n* (*n* - кількість входжень змінної *x*) формула:

a) $(...(x \to x) \to x) \to ...) \to x$ є тавтологією;

b) $(...(x \leftrightarrow x) \leftrightarrow x) \leftrightarrow ...) \leftrightarrow x$ є нейтральною;

c) $(...(x \oplus x) \oplus x) \oplus ...) \oplus x$ є виконуваною;

d) $\overline{(...(x \to \overline{x}) \to x) \to \overline{x}) \to ...)}$ є суперечністю.

1.12 На змаганнях легкоатлетів із спортивної ходьби Андрій відстав від Богдана та ще від двох спортсменів. Віктор фінішував після Дмитра, але раніше Геннадія. Дмитро випередив Богдана, але прийшов після Євгена. Яке місце займає кожний спортсмен?

1.2 ВІДНОШЕННЯ РІВНОСИЛЬНОСТІ НА БАЗІ АЛГЕБРИ ВИСЛОВЛЕНЬ

Формули $A(a_1, a_2, ... a_n)$ і $B(a_1, a_2, ... a_n)$ алгебри висловлень називаються *рівносильними* (*логічно еквівалентними*), якщо вони мають однакові значення істинності на кожному з усіх наборів логічних значень пропозиційних змінних, тобто набувають однакових значень в усіх інтерпретаціях.

Позначення: $A \equiv B$. Символ «≡» не є символом операції алгебри висловлень, а означає певне відношення між формулами; він належить не предметній мові алгебри висловлень, а метамові. Означення рівносильних формул можна записати символічно: $(A \equiv B) \Leftrightarrow (|A(a_1, a_2, ..., a_n)| = |B(a_1, a_2, ..., a_n)|)$.

Теорема 1.2.1 (ознака рівносильності формул). Дві формули A і B алгебри висловлень рівносильні тоді і тільки тоді, коли формула A↔B є тавтологією:

$$(A \equiv B) \Leftrightarrow (\models A \leftrightarrow B)$$

<u>Наслідок</u>. Відношення рівносильності між формулами алгебри висловлень має властивості:

а) рефлексивність: $A \equiv A$;

б) симетричність: якщо $A \equiv B$, то $B \equiv A$;

в) транзитивність: якщо $A \equiv B$ і $B \equiv C$, то $A \equiv C$

тобто відношення рівносильності є відношенням еквівалентності.

Теорема 1.2.2 (основні рівносильності алгебри висловлень**).** Справедливими є наступні рівносильності:

1. $a \wedge b \equiv b \wedge a$, $a \vee b \equiv b \vee a$ (*комутативні закони*);
2. $a \wedge (b \wedge c) \equiv (a \wedge b) \wedge c$, $a \vee (b \vee c) \equiv (a \vee b) \vee c$ (*асоціативні закони*);
3. $a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$, $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$ (*дистрибутивні закони*);
4. $a \wedge (b \vee a) \equiv a$, $a \vee (b \wedge a) \equiv a$ (*закони поглинання*);
5. $\overline{a \wedge b} \equiv \bar{a} \vee \bar{b}$, $\overline{a \vee b} \equiv \bar{a} \wedge \bar{b}$ (*закони де Моргана*);
6. $a \to b \equiv \bar{a} \vee b$;
7. $a \leftrightarrow b \equiv (a \to b) \wedge (b \to a)$;
8. $\bar{\bar{a}} \equiv a$ (*закон подвійного заперечення*);
9. $a \wedge a \equiv a$, $a \vee a \equiv a$ *(закони ідемпотентності)*;
10. $a \wedge 1 \equiv a$, $a \vee 1 \equiv 1$;
11. $a \wedge 0 \equiv 0$, $a \vee 0 \equiv a$;

12. $a \wedge \overline{a} \equiv 0$ *(закон протиріччя)*;

13. $a \vee \overline{a} \equiv 1$ *(закон виключеного третього);*

14. $\overline{a \wedge \overline{a}} \equiv 1$ *(закон виключення суперечності).*

Використовуючи рівносильності з теореми 1.2.2, можемо від однієї формули перейти до рівносильної їй формули. Такий перехід називається *рівносильним перетворенням* вихідної формули.

Для рівносильних перетворень та спрощень формул поряд з основними рівносильностями важливе значення має *правило заміни.*

Теорема 1.2.3 (про заміну).

Нехай $A \equiv B$ і C – деяка підформула формули A, причому $C \equiv C^*$. Тоді формула A^*, одержана з формули A заміною підформули C на C^*, рівносильна формулі B.

Відмітимо, що коли деяка формула є тавтологією, то і всяка рівносильна їй формула також є тавтологією:

$$(\models A,\ A \equiv B) \quad \Rightarrow (\models B)\,.$$

Це дозволяє виявити ще одну сферу застосування рівносильних перетворень, зокрема, для доведення тотожної істинності (чи тотожної хибності) тих або інших формул. Для цього дану формулу потрібно рівносильними перетвореннями звести до формули, яка очевидно є тавтологією (суперечністю).

Приклад 1. Довести рівносильність $a \leftrightarrow b \equiv (\overline{a} \wedge \overline{b}) \vee (a \wedge b)$.

Розв’язання: Будемо виконувати рівносильні перетворення відносно лівої частини формули: $a \leftrightarrow b \equiv (a \to b) \wedge (b \to a)$. Далі застосувавши закон $a \to b \equiv \overline{a} \vee b$ до обох імплікацій одержимо $(\overline{a} \vee b) \wedge (\overline{b} \vee a)$. Скориставшись дистрибутивними законами дістанемо $((\overline{a} \vee b) \wedge \overline{b}) \vee ((\overline{a} \vee b) \wedge a) \equiv$ $\equiv (\overline{a} \wedge \overline{b}) \vee (b \wedge \overline{b}) \vee (\overline{a} \wedge a) \vee (b \wedge a)$. Застосувавши закони $x \wedge \bar{x} = 0$, а $x \vee 0 = x$ отримаємо потрібний результат $(\overline{a} \wedge \overline{b}) \vee (a \vee b)$.

Оскільки відношення рівносильності транзитивне, то процес доведення можна записати у вигляді дедуктивного ланцюга рівносильних перетворень:

$$a \leftrightarrow b \equiv \ (a \to b) \wedge (b \to a) \ \equiv \ (\overline{a} \vee b) \wedge (\overline{b} \vee a) \ \equiv \ ((\overline{a} \vee b) \wedge \overline{b}) \vee ((\overline{a} \vee b) \wedge a) \equiv$$
$$\equiv (\overline{a} \wedge \overline{b}) \vee (b \wedge \overline{b}) \vee (\overline{a} \wedge a) \vee (b \wedge a) \ \equiv \ (\overline{a} \wedge \overline{b}) \vee (a \wedge b)\ .$$

Приклад 2. За допомогою рівносильних перетворень максимально спростити формулу та визначити тип: $(a \to b) \to ((a \vee c) \to (b \to c))$

Розв’язання.

$$(a \to b) \to ((a \vee c) \to (b \to c)) \equiv \overline{(\overline{a} \vee b)} \vee (\overline{(a \vee c)} \vee (\overline{b} \vee c)) \equiv$$

$$\equiv (a \wedge \overline{b}) \vee (\overline{a} \wedge \overline{c}) \vee (\overline{b} \vee c) \equiv (a \wedge \overline{b}) \vee (\overline{a} \wedge \overline{c}) \vee \overline{b} \vee c \equiv \overline{b} \vee \overline{a} \vee c$$

На першому кроці застосовано до всіх чотирьох імплікацій закон $a \to b \equiv \overline{a} \vee b$, на другому кроці застосовано двічі закон де Моргана, на третьому кроці опущено дужки за асоціативністю диз’юнкції, четвертий крок одержано із застосуванням законів: поглинання, дистрибутивного диз’юнкції відносно кон’юнкції, виключеного третього та $a \wedge 1 \equiv a$.

Отримана формула, як бачимо, не є тавтологією (т.я. набуває логічного значення 0 на наборі (1,1,0)) і не є суперечністю (т.я. набуває значення 1 коли a=0 або b=0 або c=1), отже це нейтральна формула.

Запитання для самоконтролю

1. Які формули алгебри висловлень називають рівносильними?
2. У чому суть ознаки рівносильності?
3. Які властивості має відношення рівносильності?
4. Якою рівносильністю записуються закони де Моргана?
5. Якою рівносильністю записуються закони поглинання?
6. Якими рівносильностями подаються комутативний, асоціативний та дистрибутивний закони алгебри висловлень?
7. Якими рівносильностями подаються закон виключення суперечності та закон виключеного третього?
8. У чому суть законів подвійного заперечення, ідемпотентності, контра позиції?
9. Сформулюйте теорему про заміну?
10. Що таке рівносильне перетворення формули? Які застосування рівносильних перетворень?

ВПРАВИ

1.13 Перевірити, чи існує рівносильність, якою визначається:

a) асоціативність дії $\to$;

b) дистрибутивність дії $\to$ відносно дії $\wedge$;

c) дистрибутивність дії $\leftrightarrow$ відносно дії $\vee$.

1.14 Довести чи спростувати рівносильність формул алгебри висловлень:

a) $a \to b \equiv \overline{(a \wedge \overline{b})}$;

b) $a \vee b \to \overline{a} \equiv \overline{a}$;

c) $(a \leftrightarrow b) \leftrightarrow c \equiv a \leftrightarrow (b \leftrightarrow c)$;

d) $a \to (b \vee c) \equiv (a \to b) \vee c$;

e) $a \vee b \vee \overline{c} \leftrightarrow a \equiv (b \to a) \wedge (\overline{a} \to c)$;

f) $a \to (b \to c) \equiv (a \wedge b) \to c$;

g) $((a \vee b) \wedge (b \to c)) \to \overline{a \to c} \equiv (a \to b) \wedge (a \to \overline{b}) \wedge a$.

1.15 Довести основні рівносильності теореми 1.2.2.

1.16 За допомогою рівносильних перетворень максимально спростити наступні формули:

a) $(\overline{b} \to \overline{c}) \wedge (b \vee (c \to \overline{a})) \leftrightarrow b \to a \wedge \overline{b}$;

b) $(\overline{(a \vee b)} \to a \wedge b \wedge c) \vee (\overline{a \wedge c})$;

c) $(a \vee (b \to c)) \wedge (a \vee b \vee c) \wedge (a \vee b \vee \overline{c})$;

d) $(a \to b) \wedge (a \vee (b \wedge c)) \wedge (a \to c) \vee \overline{c}$;

e) $(\overline{a \to b}) \vee (\overline{b \to c}) \vee c$;

f) $((a \to b) \wedge (a \vee (b \wedge c)) \wedge (a \to c)) \vee \overline{c}$.

1.17 Визначити тип формул за допомогою рівносильних перетворень:

a) $(a \to (b \to c)) \leftrightarrow (a \wedge b \to c)$;

b) $(a \vee b) \leftrightarrow \overline{b}$;

c) $\overline{a} \to c \wedge a \leftrightarrow b \wedge a \to \overline{a}$;

d) $(\overline{c} \vee a \to b \wedge c \wedge \overline{a}) \to \overline{b} \vee c$;

e) $(a \leftrightarrow b) \wedge \overline{a} \wedge b$;

f) $(b \to (a \to c)) \wedge (b \to a) \wedge b \wedge \overline{c}$.

1.18 Перетворити формули на рівносильні, що містять лише задані операції:

a) $x \vee z \wedge (y \to \overline{x})$, логічні операції: $\wedge, \neg$;

b) $(x \vee \overline{y} \to z \wedge y) \vee z$, логічні операції: $\vee, \neg$;

c) $\overline{y \wedge \overline{x}} \vee (y \wedge z)$, логічні операції: $\to, \neg$;

d) $z \to \overline{z \wedge y} \wedge x$, логічні операції: $\vee, \neg$.

1.3 НОРМАЛЬНІ ФОРМИ ФОРМУЛ АЛГЕБРИ ВИСЛОВЛЕНЬ ТА ЇХ ЗАСТОСУВАННЯ.

Кон'юнктивним (диз'юнктивним) одночленом називається кон'юнкція (диз'юнкція) пропозиційних змінних або їх заперечень (при цьому допускається повторення змінних). Кількість змінних в одночлені називають його *рангом*.

Інша назва кон'юнктивного (диз'юнктивного) одночлена – *елементарна кон'юнкція (елементарна диз'юнкція).*

Кон'юнктивні одночлени: $K_1 = a \wedge \overline{b} \wedge c$, $K_2 = a \wedge \overline{a}$, $K_3 = c$ мають ранги 3, 2, 1 відповідно.

Диз'юнктивні одночлени: $D_1 = a \vee c$, $D_2 = a \vee c \vee a \vee \overline{c}$ мають ранги 2 і 4 відповідно

Диз'юнктивною нормальною формою (кон'юнктивною нормальною формою) називається диз'юнкція кон'юнктивних одночленів (кон'юнкція диз'юнктивних одночленів). Скорочений запис ДНФ та КНФ відповідно.

Тобто ДНФ це формула виду $K_1 \vee K_2 \vee ... \vee K_p$, де всі K_i (i=1, 2,…,p) є кон'юнктивними одночленами (не обов'язково різними), а КНФ - $D_1 \wedge D_2 . \wedge .. \wedge D_q$, де всі D_i, i=1, 2,…,q є диз'юнктивними одночленами (не обов'язково різними).

Будь-яка формула алгебри висловлень може бути зведена до рівносильних їй ДНФ та КНФ (причому неоднозначно).

Приклад 1. Побудувати КНФ та ДНФ, рівносильні формулі: $A = (x \wedge \overline{y} \to z) \to \overline{x}$.

Розв'язання.

Перший крок – будуємо формулу, рівносильну до А, що містить лише операції $\wedge, \vee, \neg$:

$$(x \wedge \overline{y} \to z) \to \overline{x} \equiv (\overline{x \wedge \overline{y}} \vee z) \to \overline{x} \equiv \overline{(\overline{x \wedge \overline{y}} \vee z)} \vee \overline{x}.$$

Другий крок – знайдемо рівносильну формулу, в якій знак заперечення стоїть лише біля змінних: $\overline{(\overline{x \wedge \overline{y}} \vee z)} \vee \overline{x} \equiv (\overline{\overline{x \wedge \overline{y}}} \wedge \overline{z}) \vee \overline{x} \equiv (x \wedge \overline{y} \wedge \overline{z}) \vee \overline{x}$. Одержана формула є ДНФ(А), що складається з двох кон'юнктивних одночленів: $K_1 = (x \wedge \overline{y} \wedge \overline{z})$ та $K_2 = \overline{x}$. Щоб дістати КНФ(А) треба застосувати дистрибутивний закон диз'юнкції відносно кон'юнкції до ДНФ(А):

$(x \wedge \overline{y} \wedge \overline{z}) \vee \overline{x} \equiv (x \vee \overline{x}) \wedge (\overline{y} \vee \overline{x}) \wedge (\overline{z} \vee \overline{x})$. Ми отримали КНФ(А), яка містить три диз'юнктивні одночлени: $D_1 = x \vee \overline{x}$, $D_2 = \overline{y} \vee \overline{x}$, $D_3 = \overline{z} \vee \overline{x}$.

Отже, КНФ(А)= $(x \vee \overline{x}) \wedge (\overline{y} \vee \overline{x}) \wedge (\overline{z} \vee \overline{x})$, ДНФ(А)= $(x \wedge \overline{y} \wedge \overline{z}) \vee \overline{x}$.

Одним із застосувань нормальних форм є встановлення типу формул алгебри висловлень (спосіб розв'язання проблеми вирішення в алгебрі висловлень).

Теорема 1.3.1. Формула алгебри висловлень є тавтологією тоді і тільки тоді, коли кожний диз'юнктивний одночлен рівносильної їй КНФ, містить хоча б одну пропозиційну змінну разом з її запереченням.

Теорема 1.3.2. Формула алгебри висловлень є суперечністю тоді і тільки тоді, коли кожен кон'юнктивний одночлен рівносильної їй ДНФ містить хоча б одну пропозиційну змінну разом з її запереченням.

Застосувавши теореми 1.3.1 та 1.3.2 до формули $A = (x \wedge \overline{y} \to z) \to \overline{x}$ попереднього приклада визначимо її тип. Оскільки $D_2 = \overline{y} \vee \overline{x}$, $D_3 = \overline{z} \vee \overline{x}$ в КНФ(А) не містять деяку пропозиційну змінну разом з її запереченням, то А не тавтологія. Аналогічно, в ДНФ(А) кон'юнктивні одночлени $K_1 = x \wedge \overline{y} \wedge \overline{z}$ і $K_2 = \overline{x}$ не містять змінну разом з її запереченням, отже формула А не є суперечністю. Значить формула А нейтральна.

Серед багатьох ДНФ (КНФ), які рівносильні заданій формулі алгебри висловлень, існує унікальна нормальна форма, яка визначається однозначно (з точністю до порядку запису одночленів): це так звана досконала диз'юнктивна (кон'юнктивна) нормальна форма.

Одночлен (кон'юнктивний або диз'юнктивний) називається *правильним*, якщо в нього кожна змінна(або її заперечення), зустрічається не більше, ніж один раз.

Одночлен (кон'юнктивний або диз'юнктивний) називається *повним*, відносно набору змінних, якщо кожна змінна входить в нього хоч раз.

Нормальна форма називається *досконалою*, якщо всі одночлени, що в неї входять, є правильними, повними і різними одночасно. Досконала ДНФ (КНФ) позначається ДДНФ (ДКНФ).

Будувати досконалі форми для заданої формули можна за допомогою рівносильних перетворень або використовуючи таблицю істинності. Проілюструємо це на прикладі.

Приклад 2. Для формули А=$a \vee \bar{b} \leftrightarrow c$ знайти ДКНФ(А).

Спосіб рівносильних перетворень (алгебраїчний спосіб).

Перший крок - побудова КНФ(А):

$$A = a \vee \bar{b} \leftrightarrow c \equiv \left(\left(a \vee \bar{b}\right) \to c\right) \wedge \left(c \to \left(a \vee \bar{b}\right)\right) \equiv \left(\left(\overline{a \vee \bar{b}}\right) \vee c\right) \wedge \left(\bar{c} \vee \left(a \vee \bar{b}\right)\right)$$
$$\equiv \left(\left(\bar{a} \wedge b\right) \vee c\right) \wedge \left(\bar{c} \vee a \vee \bar{b}\right) \equiv \left(\left(\bar{a} \wedge b\right) \vee c\right) \wedge \left(\bar{c} \vee a \vee \bar{b}\right) \equiv$$
$$\equiv \left(\bar{a} \vee c\right) \wedge \left(b \vee c\right) \wedge \left(\bar{c} \vee a \vee \bar{b}\right).$$

Отримана КНФ(А) не є досконалою, тому що перші два диз’юнктивні одночлени не є повними.

Другий крок – перетворення кожного неповного (неправильного) одночлена в повний (правильний). Для цього використаємо наступні рівносильності: $x \vee 0 = x, \quad x \wedge \bar{x} = 0$.

$$D_1 \equiv \left(\bar{a} \vee c\right) \equiv \left(\bar{a} \vee c \vee 0\right) \equiv \left(\bar{a} \vee c \vee \left(b \wedge \bar{b}\right)\right) \equiv \left(\bar{a} \vee c \vee b\right) \wedge \left(\bar{a} \vee c \vee \bar{b}\right),$$

$$D_2 \equiv \left(b \vee c\right) \equiv \left(b \vee c \vee 0\right) \equiv \left(b \vee c \vee \left(a \wedge \bar{a}\right)\right) \equiv \left(b \vee c \vee a\right) \wedge \left(b \vee c \vee \bar{a}\right).$$

Третій крок – записати ДКНФ(А), де жоден одночлен не повторюється і змінні в одночленах розташовані в певному порядку:

$$A \equiv \left(\bar{a} \vee c \vee b\right) \wedge \left(\bar{a} \vee c \vee \bar{b}\right) \wedge \left(b \vee c \vee a\right) \wedge \left(b \vee c \vee \bar{a}\right) \wedge \left(\bar{c} \vee a \vee \bar{b}\right)$$

$\equiv \left(\bar{a} \vee b \vee c\right) \wedge \left(\bar{a} \vee \bar{b} \vee c\right) \wedge \left(a \vee b \vee c\right) \wedge \left(a \vee b \vee \bar{c}\right)$ Отримали ДКНФ, рівносильну заданій формулі. Таким чином,

ДКНФ(А)= $\equiv \left(\bar{a} \vee b \vee c\right) \wedge \left(\bar{a} \vee \bar{b} \vee c\right) \wedge \left(a \vee b \vee c\right) \wedge \left(a \vee b \vee \bar{c}\right).$

Табличний спосіб.

Перший крок - побудова таблиці істинності формули

Другий крок – побудова конституент нуля відповідно всіх тих наборів змінних, на яких формула набуває хибного значення.

Конституентою нуля (макстермом) називають досконалий диз’юнктивний одночлен, який набуває значення 0 тільки на відповідному наборі логічних змінних: $D = x_1^\delta \vee x_2^\delta \vee ... \vee x_n^\delta$, де $x_i^\delta = \begin{cases} x_i, & |x_i| = 0 \\ \overline{x_i}, & |x_i| = 1 \end{cases}$

Третій крок - запис ДКНФ, як кон’юнкції тих конституент нуля, які перетворюються на нуль на тих наборах значень змінних, що й задана формула.

Будуємо таблицю істинності формули

$\|a\|$	$\|b\|$	$\|c\|$	$\|\bar{b}\|$	$\|a \vee \bar{b}\|$	$\|a \vee \bar{b} \leftrightarrow c\|$	**Конституенти**	**Конституенти**

						нуля	одиниці
0	0	0	1	1	0	$D_1 = a \vee b \vee c$	
0	0	1	1	1	1		$K_1 = \bar{a} \wedge \bar{b} \wedge c$
0	1	0	0	0	1		$K_2 = \bar{a} \wedge b \wedge \bar{c}$
0	1	1	0	0	0	$D_2 = a \vee \bar{b} \vee \bar{c}$	
1	0	0	1	1	0	$D_3 = \bar{a} \vee b \vee c$	
1	0	1	1	1	1		$K_3 = a \wedge \bar{b} \wedge c$
1	1	0	0	1	0	$D_4 = \bar{a} \vee \bar{b} \vee c$	
1	1	1	0	1	1		$K_4 = a \wedge b \wedge c$

ДКНФ заданої формули утворюється кон'юнкцією конституент нуля , отже:

ДКНФ(А) $\equiv (a \vee b \vee c) \wedge (a \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee c) \wedge (\bar{a} \vee \bar{b} \vee c)$.

Аналогічно, ДДНФ(А) визначається диз'юнкцією конституент одиниці. *Конституентою одиниці* (мінтермом) називають досконалий кон'юнктивний одночлен, який набуває значення 1 тільки на відповідному наборі логічних змінних: $K = x_1^{\delta} \wedge x_2^{\delta} \wedge \ldots \wedge x_n^{\delta}$, де $x_i^{\delta} = \begin{cases} x_i, & |x_i| = 1 \\ \overline{x_i}, & |x_i| = 0 \end{cases}$.

ДДНФ(А) $\equiv (\bar{a} \wedge \bar{b} \wedge c) \vee (\bar{a} \wedge b \wedge \bar{c}) \vee (a \wedge \bar{b} \wedge c) \vee (a \wedge b \wedge c)$.

Теорема 1.3.3 Формула алгебри висловлень $A(a_1, a_2, \ldots, a_n)$ є тавтологією тоді і тільки тоді, коли рівносильна їй ДДНФ(А) містить 2^n кон'юнктивних одночленів.

Теорема 1.3.4 Формула алгебри висловлень $A(a_1, a_2, \ldots, a_n)$ є суперечністю тоді і тільки тоді, коли рівносильна їй ДКНФ(А) містить 2^n диз'юнктивних одночленів.

Запитання для самоконтролю

1. Яку формулу алгебри висловлень називають кон'юнктивним (диз'юнктивним) одночленом або елементарною кон'юнкцією (елементарною диз'юнкцією)?
2. Що називається диз'юнктивною нормальною формою (ДНФ)?
3. Що називається кон'юнктивною нормальною формою (КНФ)?
4. Чи кожна формула алгебри висловлень може бути зведена до рівносильних їй ДНФ та КНФ?
5. Який алгоритм побудови нормальних форм?

6. Як застосовуються нормальні форми до проблеми вирішення алгебри висловлень?
7. Який одночлен (кон'юнктивний або диз'юнктивний) називається правильним?
8. Який одночлен називається повним відносно набору змінних?
9. Яка нормальна форма (кон'юнктивна або диз'юнктивна) називається досконалою (ДДНФ, ДКНФ)?
10. Який алгоритм побудови досконалих форм способом рівносильних перетворень?
11. Який алгоритм побудови досконалих форм способом табличним способом?
12. Яку формулу алгебри висловлень називають конституентою нуля (макстермом)?
13. Яку формулу алгебри висловлень називають конституантою одиниці (мін термом)?
14. Як застосовуються досконалі нормальні форми до проблеми вирішення алгебри висловлень?

ВПРАВИ

1.19 Побудувати ДНФ та КНФ, рівносильні формулам:

a) $a \vee b \to \bar{a} \wedge b$;

b) $\overline{a \wedge b} \vee (a \to b)$;

c) $(a \vee b) \wedge c \to \bar{a} \vee c$;

d) $(a \to b) \wedge (b \to a) \leftrightarrow (a \leftrightarrow b)$;

e) $(a \wedge c \to b) \to (\bar{a} \leftrightarrow c)$;

f) $\overline{(a \leftrightarrow b)}\overline{\leftrightarrow c}$;

g) $(b \to c) \to ((a \to c) \to (a \vee b \to c))$.

1.20 Зведенням формули до ДНФ та КНФ визначити тип формули:

a) $a \wedge (\overline{b \to a})$;

b) $a \vee b \to (a \leftrightarrow b)$;

c) $(c \to c \wedge b \to a) \vee \overline{\bar{b} \wedge c}$;

d) $\overline{a \vee \bar{b} \to c \wedge b} \to \bar{a} \vee c$;

e) $(a \leftrightarrow c \vee c \wedge b) \to (a \to c)$;

f) $((a \wedge b \wedge c) \vee (c \wedge b) \rightarrow a) \vee (b \wedge c)$;

g) $\left(a \wedge \left(b \wedge \left(\overline{c \rightarrow d}\right)\right) \leftrightarrow (c \rightarrow d)\right) \vee b$.

1.21 Перетворивши дані досконалі форми, знайти формули, які їм рівносильні та містять найменшу кількість операцій:

a) $a \wedge b \vee a \wedge \bar{b} \vee \bar{a} \wedge b$;

b) $(a \vee \bar{b}) \wedge (\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b})$;

c) $a \wedge b \wedge c \vee \bar{a} \wedge b \wedge c \vee a \wedge \bar{b} \wedge c$;

d) $(a \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee c) \wedge (a \vee \bar{b} \vee \bar{c})$.

1.22 Звести формули до ДДНФ та ДКНФ двома способами – табличним та алгебраїчним:

a) $a \leftrightarrow \bar{a} \wedge b$;

b) $(a \rightarrow b \wedge c) \rightarrow a \vee c$;

c) $\left(\overline{b \wedge a} \rightarrow \overline{c \wedge b \rightarrow a}\right) \vee \left(\bar{b} \wedge c\right)$;

d) $\left(\overline{b \rightarrow a} \rightarrow c \wedge b \rightarrow a\right) \vee (b \wedge c)$;

e) $\left(\overline{a \rightarrow b \wedge a \wedge c}\right) \vee (a \wedge c \wedge b)$;

f) $\left(a \wedge \bar{b} \rightarrow c \wedge b\right) \rightarrow c \vee (b \wedge c)$;

g) $\left(a \vee c \wedge b \rightarrow \bar{a}\right) \wedge b \vee (b \wedge c)$.

1.23 Знаючи ДКНФ(A) і ДКНФ(B), побудувати ДКНФ($A \vee B$).

1.24 Знаючи ДДНФ(A) і ДДНФ(B), побудувати ДДНФ($A \rightarrow B$) та ДКНФ($A \rightarrow B$).

1.25 Знайти ранг ДКНФ, яка рівносильна формулі:

a) $x_1 \rightarrow (x_2 \rightarrow \ldots \rightarrow (x_{n-1} \rightarrow x_n)\ldots)$;

б) $x_1 \leftrightarrow x_2 \leftrightarrow \ldots \leftrightarrow x_n$.

1.4 БУЛЕВІ ФУНКЦІЇ. ФУНКЦІОНАЛЬНА ПОВНОТА. ПОЛІНОМИ ЖЕГАЛКІНА

Впорядкований набір $\bar{a} = (\alpha_1, \alpha_2, \ldots \alpha_n)$ логічних значень $\alpha_i \in \{0,1\}$ називається *булевим вектором*, а число n – довжиною мулевого вектора $\bar{a}$. Тоді множина $\{0,1\}^n = \{(\alpha_1, \ldots \alpha_n) / \alpha_s \in \{0,1\}, i = \overline{1,n}\}$ є множиною всіх булевих векторів довжини n.

Функція $f(x_1, x_2, \ldots, x_n)$ від n- змінних (n≥1) називається *булевою*, якщо кожний її аргумент і сама функція приймають значення на множині $\{0,1\}$:

$$f:\{0,1\}^n \to \{0,1\}.$$

Для кожного висловлення, одержаного з n простих висловлень існує булева функція від n змінних, значення якої при відповідних значеннях змінних збігається з оцінкою висловлення – це функція істинності даного висловлення.

В обчислювальній техніці булеві функції використовують для опису алгоритмів та дискретних пристроїв. Дискретні пристрої призначаються для перетворення дискретної інформації, що розкладається на елементарні одиниці – біти, які в пристроях реалізуються сигналами, що описуються двійковими (булевими) змінними.

Булеву n-місну функцію $f(x_1, x_2, ..., x_n)$ можна задати:

а) безпосередньо таблицею, на якій точно визначено, яке значення f відповідає кожному впорядкованій набору значень аргументів $x_1, ..., x_n$ (таблицею істинності);

б) формулою, в якій задано операції, що їх треба виконати над значеннями аргументів $x_1, ..., x_n$, щоб дістати значення функції, наприклад, $f(x_1, x_2, x_3) = x_1 \oplus x_2 \downarrow \overline{x_3 \to x_1}$ (аналітичний спосіб);

в) словесним описом чи характеристичною властивістю, наприклад, 1) «функція $f(x_1, x_2, ..., x_n)$ набуває значення 1 тільки тоді, коли всі аргументи набувають однакових значень» або 2) $f(x_1, x_2, x_3) = \begin{cases} 1, & x_1 = x_3 \\ 0, & x_1 \neq x_3 \end{cases}$ (вербальний спосіб).

Постає питання про кількість булевих функцій від заданого числа змінних.

Теорема 1.4.1. Число всіх наборів значень аргументів булевої функції $f(x_1, x_2, ..., x_n)$ дорівнює 2^n, а число всіх булевих функцій від n змінних дорівнює 2^{2^n}

1. $n = 1$. Задамо табличним способом всі булеві функції від однієї змінної.

x	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

Аналітично вони записуються так: f_0(x) = 0, f_1(x) = x, f_2(x) = $\bar{x}$, f_3(x) =1.

2. n = 2. Знову задамо табличним способом всі булеві функції від двох змінних. Одержимо:

x	y	g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}	g_{11}	g_{12}	g_{13}	g_{14}	g_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
		0	$\wedge$	$\overline{\rightarrow}$	x	$\overline{\leftarrow}$	y	$\oplus$	$\vee$	$\overline{\vee}$	$\leftrightarrow$	$\overline{y}$	$\leftarrow$	$\overline{x}$	$\rightarrow$	$\overline{\wedge}$	1

Отже, є 16 булевих функцій від двох змінних. Більшість із них має свої назви і спеціальні позначення, а саме:

$g_0(x,y) = 0$ – тотожний нуль;

$g_{15}(x,y) = 1$ – тотожна одиниця;

$g_1(x,y) = x \wedge y$ – кон'юнкція (множення) x та y;

$g_{14}(x,y) = \overline{x \wedge y} = x|y$ – заперечення (інверсія) кон'юнкції або штрих Шеффера;

$g_7(x,y) = x \vee y$ – диз'юнкція (додавання) x та y;

$g_8(x,y) = \overline{x \vee y} = x \downarrow y$ – заперечення (інверсія) диз'юнкції, або стрілка Пірса;

$g_{13}(x,y) = x \rightarrow y$ – імплікація x та y;

$g_2(x,y) = \overline{x \rightarrow y}$ - заперечення (інверсія) імплікації x та y;

$g_{11}(x,y) = y \rightarrow x$ – імплікація y та x;

$g_4(x,y) = \overline{y \rightarrow x}$ - заперечення (інверсія) імплікації y та x;

$g_9(x,y) = x \leftrightarrow y$ – еквіваленція x та y;

$g_6(x,y) = \overline{x \leftrightarrow y} = x \oplus y$ - сума Жегалкіна або додавання за mod2 (заперечення еквіваленції x та y);

$g_3(x,y) = x$;

$g_{12}(x,y) = \overline{x}$ - інверсія x;

$g_5(x,y) = y$;

$g_{10}(x,y) = \overline{y}$ - інверсія y.

Кожній формулі алгебри висловлень відповідає булева функція (функція істинності). Чи має місце обернене твердження, тобто, чи для кожної булевої функції існує формула алгебри висловлень, яка реалізує цю функцію?

Теорема 1.4.2. Кожна булева функція задається формулою алгебри висловлень, яка містить символи не більш, ніж трьох логічних операцій – кон'юнкції, диз'юнкції, заперечення.

Дійсно, якщо булева функція $f(x_1, x_2, ..., x_n) = 0$ – тотожній нуль, то її можна представити, наприклад, у вигляді $f(x_1, x_2, ..., x_n) = x_1 \wedge \overline{x_1}$. Якщо $f(x_1, x_2, ..., x_n)$ набуває значення 1 на деяких наборах (нехай m – кількість одиниць булевої функції), то занумерувавши ці набори аргументів будуємо конституенти одиниці: $K_j = \wedge_{i=1}^{n} x_i^{\sigma}$, де $x_i^{\sigma} = \begin{cases} x_i, \text{якщо } |x_i| = 1, \\ \rceil x_i, \text{ якщо } |x_i| = 0. \end{cases}$ $(1 \le j \le m, 1 \le m \le 2^n)$.

Тоді формулою, що реалізує f буде диз'юнкція всіх K_j:

$f(x_1, x_2, ..., x_n) = \vee_{j=1}^{m} K_j$, причому вона є досконалою диз'юнктивною нормальною формою.

Не кожна система операцій алгебри висловлень має властивість, визначену теоремою 1.4.2 для системи $\{\neg, \wedge, \vee\}$. Наприклад, через →, ↔ не можна зобразити всі булеві функції, оскільки не існує такого зображення для булевої функції f^*, яка мала б значення 0 на наборі (1, 1, …, 1), що складається з одних лише одиниць. На цьому наборі значення кожної з операцій →, ↔, а отже, і їх суперпозицій, дорівнює 1.

Система операцій алгебри висловлень (система булевих функцій) називається *функціонально повною*, якщо довільну булеву функцію можна зобразити формулою, що містить лише логічні операції (є суперпозицією функцій) даної системи.

Теорема 1.4.3. Наступні системи є функціонально повними:

1) $\{\neg, \wedge, \vee\}$; 2) $\{\neg, \wedge\}$; 3) $\{\neg, \vee\}$; 4) $\{\neg, \rightarrow\}$; 5) $\{0, \rightarrow\}$; 6) $\{\oplus, \wedge, 1\}$; 7) $\{\mid\}$; 8) $\{\downarrow\}$.

Задання булевої функції формулою, що містить лише функції системи $\{\oplus, \wedge, 1\}$ називають *поліномом Жегалкіна*. *Канонічним видом* полінома Жегалкіна є наступне представлення булевої функції:

$$f(x_1, x_2, ..., x_n) = k_0 \oplus k_1 \wedge x_1 \oplus k_2 \wedge x_2 \oplus k_{12} \wedge x_1 \wedge x_2 \oplus ... \oplus k_{i...i_t} \wedge x_i \wedge ... \wedge x_{i_t} \oplus ... \oplus$$
$$\oplus k_{12...n} \wedge x_1 \wedge x_2 \wedge ... \wedge x_n$$

або $f(x_1, x_2, ..., x_n) = k_0 + k_1 x_1 + k_2 x_2 + k_{12} x_1 x_2 + ... + k_{i...i_t} x_i ... x_{i_t} + ... + k_{12...n} x_1 x_2 ... x_n$, де коефіцієнти $k_{i...i_t} \in \{0,1\}$ і мають узагальнені індекси, що вказують на їх належність до кон'юнкції (добутку) аргументів $x_i \wedge \ldots \wedge x_{i_t}$ $(x_i \ldots x_{i_t})$. Будувати поліноми Жегалкіна можна за допомогою: 1) методу невизначених коефіцієнтів, 2) рівносильних перетворень, 3) досконалої диз'юнктивної нормальної форми (табличний спосіб).

Приклад 1: Знайти канонічний вид полінома Жегалкіна для заданої функції: $f(x_1, x_2) = x_1 \vee x_2$.

Розв'язання. (спосіб 1) Використаємо, спочатку, метод невизначених коефіцієнтів. Канонічний вид полінома для функції двох змінних має наступний вигляд: $k_0 + k_1 x_1 + k_2 x_2 + k_{12} x_1 x_2$, де $k_0, k_1, k_2, k_{12} \in \{0,1\}$. Будемо послідовно підставляти набори значень аргументів в задану формулу та полінном:

$f(0,0) = 0 \vee 0 = 0$ і $k_0 + k_1 0 + k_2 0 + k_{12} 0 \cdot 0 = k_0$, отже $k_0 = 0$;

$f(0,1) = 0 \vee 1 = 1$ і $k_0 + k_1 0 + k_2 1 + k_{12} 0 \cdot 1 = k_0 + k_2$, отже $k_2 = 1$;

$f(1,0) = 1 \vee 0 = 1$ і $k_0 + k_1 1 + k_2 0 + k_{12} 1 \cdot 0 = k_0 + k_1$, отже $k_1 = 1$;

$f(1,1) = 1 \vee 1 = 1$ і $k_0 + k_1 1 + k_2 1 + k_{12} 1 \cdot 1 = k_0 + k_1 + k_2 + k_{12}$, отже $1 + 1 + k_{12} = 1; \quad k_{12} = 1$;

Підставивши знайдені значення коефіцієнтів одержимо: $x_1 \vee x_2 = x_1 + x_2 + x_1 x_2$.

(спосіб 2) Застосуємо рівносильні перетворення до заданої функції:

$$x_1 \vee x_2 \equiv \overline{\overline{x_1 \vee x_2}} \equiv \overline{\overline{x_1} \wedge \overline{x_2}} \equiv \overline{(x_1 \oplus 1) \wedge (x_2 \oplus 1)} \equiv ((x_1 \oplus 1) \wedge (x_2 \oplus 1)) \oplus 1 \equiv$$
$$\equiv (x_1 \wedge x_2 \oplus x_1 \wedge 1 \oplus 1 \wedge x_2 \oplus 1 \wedge 1) \oplus 1 \equiv x_1 \wedge x_2 \oplus x_1 \oplus x_2 \oplus 1 \oplus 1 \equiv x_1 \wedge x_2 \oplus x_1 \oplus x_2$$

або $x_1 \vee x_2 \equiv x_1 \wedge x_2 \oplus x_1 \oplus x_2 = x_1 x_2 + x_1 + x_2$.

Приклад 2: Знайти канонічний вид полінома Жегалкіна для заданої формули: $(a \leftrightarrow c \vee b) \wedge c$ використовуючи ДДНФ.

Для побудови полінома Жегалкіна функції $f(x_1, x_2, \ldots, x_n)$, яка не є тотожнім нулем, за допомогою ДДНФ треба:

1) знайти ДДНФ(*f*),
2) замінити $\vee$ на $\oplus$ (+),
3) замінити $\overline{x_i}$ на $(x_i \oplus 1)$,
4) спростити, використовуючи закони:

$x \wedge 1 \equiv x, \ x \oplus 0 \equiv x, \ 1 \oplus 1 \equiv 0, \ x \oplus x \equiv 0, \ x \wedge (y \oplus z) \equiv x \wedge y \oplus x \wedge z$.

Спочатку побудуємо ДДНФ. Для цього побудуємо таблицю істинності для даної формули: A=$(a \leftrightarrow c \vee b) \wedge c$.

$\lvert a \rvert$	$\lvert b \rvert$	$\lvert c \rvert$	$\lvert c \vee b \rvert$	$\lvert a \leftrightarrow c \vee b \rvert$	$\lvert (a \leftrightarrow c \vee b) \wedge c \rvert$
0	0	0	0	1	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	1	1	1

Отримаємо два кон'юнктивні одночлени: $K_1 = a \wedge \bar{b} \wedge c$, $K_2 = a \wedge b \wedge c$.

ДДНФ нашої формули має такий вигляд: A=$(a \wedge \bar{b} \wedge c) \vee (a \wedge b \wedge c)$.

Знак диз'юнкції ($\vee$) замінимо на знак додавання за mod2 ($\oplus$) і використаємо рівносильність $\bar{a} \equiv a \oplus 1$. Отримаємо:

$$(a \wedge \bar{b} \wedge c) \vee (a \wedge b \wedge c) \equiv (a \wedge (b \oplus 1) \wedge c) \oplus (a \wedge b \wedge c) \equiv$$
$$\equiv (a \wedge b \wedge c \oplus a \wedge c) \oplus (a \wedge b \wedge c) \equiv$$

Опустимо дужки, знак кон'юнкції ($\wedge$) замінимо на знак множення ($\cdot$):

$$\equiv abc \oplus ac \oplus abc \equiv$$

Скористаємось наступними властивостями виключного додавання $x \oplus x = 0$, $x \oplus 0 = x$. Отримаємо: A$\equiv ac$.

Виділяють п'ять класів булевих функцій.

Множина булевих функцій:

$T_0 = \{f(x_1, x_2, ..., x_n) / f(0,0,...,0) = 0\}$ називається класом функцій, що *зберігають нуль*.

$T_1 = \{f(x_1, x_2, ..., x_n) / f(1,1,...,1) = 1\}$ називається класом функцій, що *зберігають одиницю*.

$S = \{f(x_1, x_2, ..., x_n) / f(\overline{x_1}, ..., \overline{x_n}) = \overline{f(x_1, ..., x_n)}\}$ називається класом *самодвоїстих функцій*.

$M = \{f(x_1, x_2, ..., x_n) / f(\alpha_1, \alpha_2, ..., \alpha_n) \le f(\beta_1, \beta_2, ..., \beta_n), \alpha_i \le \beta_i, i = \overline{1,n}\}$ називається класом *монотонних функцій*.

$$L = \left\{ f(x_1, x_2, ..., x_n) / f = k_0 \oplus k_1 x_1 \oplus k_2 x_2 \oplus ... \oplus k_n x_n,\ k_i \in \{0,1\},\ i = \overline{1,n} \right\}$$

називається класом *лінійних функцій*.

Теорема 1.4.4. (*Критерій Е. Поста*) Система булевих функцій буде функціонально повною тоді і лише тоді, коли вона не міститься в жодному з класів T_0, T_1, S, M, L.

Запитання для самоконтролю

1. Що називається булевим вектором? Скільки різних булевих векторів довжини n існує?
2. Що називається булевою функцією від n змінних? Як вони задаються?
3. Скільки існує різних булевих функцій від n змінних?
4. Які булеві функції однієї змінної визнаєте?
5. Які булеві функції двох змінних існують?
6. Як визначаються штрих Шеффера, стрілка Пірса, сума Жегалкіна?
7. Який зв'язок між булевими функціями та формулами алгебри висловлень?
8. Яка система булевих функцій називається функціонально повною?
9. Які функціонально повні системи ви знаєте?
10. Яку булеву функцію називають поліномом Жегалкіна?
11. Що таке канонічний вид полінома Жегалкіна? Які існують методи для побудови полінома Жегалкіна?
12. Які існують класи булевих функцій?
13. Як визначаються класи функцій, що зберігають нуль (зберігають одиницю)? Наведіть приклади функцій, що відносяться до цих класів.
14. Як визначаються клас монотонних функцій, клас само двоїстих функцій та клас лінійних функцій? Наведіть приклади булевих функцій, що відносяться до цих класів.
15. В чому суть критерія Поста?

ВПРАВИ

1.26 Знайти значення булевої функції (таблицю істинності), заданої логічною формулою та деяку рівносильну формулу алгебри висловлень:

a) $(x \wedge z \leftrightarrow (z \oplus y) \rightarrow x) \downarrow \overline{y} \wedge z$;

b) $(z \leftarrow (x \oplus (z \downarrow y)) \rightarrow x) \leftarrow (y \wedge \overline{z})$;

c) $(x \leftrightarrow z) \vee (z \downarrow y) \leftrightarrow \overline{(x \leftarrow (x \oplus z) \vee z)}$;

d) $(z \downarrow (\overline{x \leftrightarrow z}) \oplus y \rightarrow x) \downarrow (\overline{y} \wedge z)$;

e) $\left(\overline{(x \leftarrow y)} \oplus (z / y) \rightarrow x\right) \downarrow \left(\overline{y \wedge z}\right)$;

f) $(x \wedge z \leftrightarrow (z / y) \rightarrow x) \downarrow \overline{(y \wedge z)}$;

g) $\left(\overline{x \leftarrow (z \wedge y)} \leftrightarrow (x \oplus (x \downarrow z) \oplus z)\right)$;

h) $(z \leftarrow (x \leftrightarrow z) \oplus y \rightarrow x) \downarrow (\bar{y} \wedge z)$;

i) $(z \oplus x) \downarrow (z \wedge y \rightarrow x) \vee \overline{(\bar{y} \leftarrow z)}$;

j) $\overline{(y \leftarrow (z \wedge y)) \downarrow (x \leftrightarrow y) \oplus z \oplus y}$.

1.27 Довести функціональну повноту наступних систем операцій алгебри висловлень не використовуючи теорему Поста: а) $\{\neg, \vee\}$; б) $\{\neg, \rightarrow\}$; в) $\{0, \rightarrow\}$; г) $\{\oplus, \wedge, 1\}$; д) $\{\,|\,\}$; е) $\{\downarrow\}$.

1.28 Визначити, які з булевих функцій двох змінних належать до класів T_0, T_1, L, S, M. Відповідь оформити у вигляді таблиці. Обґрунтувати функціональну повноту систем булевих функції прикладу 1.27 опираючись на теорему Е.Поста.

1.29 Довести, що кожна з наступних систем операцій алгебри висловлень не є функціонально повною : а) $\{\vee, \wedge, 1\}$; б) $\{\neg, \leftrightarrow\}$; в) $\{\oplus, \vee\}$; г) $\{\rightarrow, \leftrightarrow\}$.

1.30 Які формули алгебри висловлень подаються такими поліномами Жегалкіна:

a) $x \wedge y \oplus x \oplus 1$;

b) $x \wedge y \oplus x \oplus y$;

c) $x \wedge y \wedge z \oplus x \oplus 1$;

d) $x \wedge y \wedge z \oplus x \oplus y \oplus z$.

1.31 Знайти канонічний вид поліномів Жегалкіна, що зображають наступні функції Буля використовуючи метод невизначених коефіцієнтів:

a) $\bar{x} \leftrightarrow x \wedge y$;

b) $z \rightarrow x \rightarrow \overline{x \vee z}$;

c) $(y \rightarrow x) \vee x \leftrightarrow z$;

d) $\overline{(z \leftrightarrow x \vee y)} \vee z$.

1.32. Знайти канонічний вид поліномів Жегалкіна, що зображають наступні функції Буля використовуючи метод рівносильних перетворень:

a) $\left(\overline{x \vee y} \rightarrow (x \wedge y \wedge z)\right) \vee \overline{(x \wedge z)}$;

b) $\overline{(z \to x)} \vee \overline{(y \vee x)} \vee x$;

c) $(y \to x) \vee \overline{(x \leftrightarrow z)}$;

d) $\overline{(z \to x \vee y)} \vee z$.

1.33 Знайти канонічний вид поліномів Жегалкіна, що зображають наступні функції Буля за допомогою ДДНФ:

a) $\left(\overline{x \vee y} \leftrightarrow (x \wedge y \wedge z)\right) \to z$;

b) $\overline{(z \to x)} \wedge \overline{(y \vee x)} \leftrightarrow x$;

c) $(y \to x) \vee x \leftrightarrow z$;

d) $\overline{(z \to x \vee y)} \wedge \overline{(x \to y)}$.

1.34 Потрійною стрілкою Лукасевича назвемо логічну дію $\uparrow_3$ яка визначається так: висловлення $\uparrow_3 (x_1, x_2, x_3)$ істинне тоді й тільки тоді, коли x_1, x_2, x_3 - хибні висловлення. Чи буде система $\{\uparrow_3\}$ повною?

1.5 ТЕХНІЧНІ ЗАСТОСУВАННЯ АЛГЕБРИ БУЛЕВИХ ФУНКЦІЙ

Комбінаційна схема (КС) – технічний пристрій, який служить для перетворення дискретних повідомлень і має n входів і m виходів і такий, що сукупність вихідних сигналів в кожний дискретний момент часу однозначно визначається вхідними сигналами, які подаються на вхід в той самий момент часу. Сигнали, які подаються на входи і знімаються з виходів, можуть приймати тільки значення “1” і “0”. Такими сигналами можуть, наприклад, бути високий і низький рівні напруги. Отже КС перетворює деяке n-буквене слово в двійковому алфавіті в m-буквенне слово в тому самому алфавіті.

КС найчастіше будують з елементарних логічних елементів, які реалізують найпростіші логічні операції.

Умовні позначення базових логічних елементів показано нижче.

НЕ (¬) **І (Λ)** **АБО (V)** ⊕ **І-НЕ (|)** **АБО-НЕ (↓)**

Маючи запас таких елементів, можна будувати більш складні схеми, під’єднуючи виходи одних елементів до входів інших. Якщо при таких з'єднаннях уникати виникнення замкнутих контурів (наприклад, під'єднання виходу елемента на один з його власних входів), то отримуємо комбінаційну схему. Такі схеми знаходяться в однозначній відповідності з формулами булевої алгебри, так, що з їх допомогою може бути виражена будь-яка система булевих функцій.

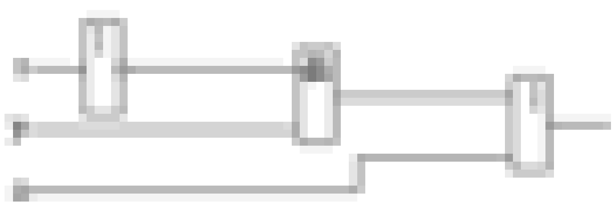
Логічний елемент, який реалізує операцію заперечення, називається елементом «НЕ», або *інвентором*. Він має один вхід, на який подається сигнал, що відповідає булевій змінній x (якщо є сигнал, то x=1, якщо немає сигналу - x=0), і один вихід, на якому виникає сигнал $\bar{x}$. Логічний елемент, який реалізує операцію кон'юнкції, називається елементом «І», або *збігом*. Логічний елемент, який реалізує операцію диз'юнкції, називається елементом «АБО» (*відокремленням*). Логічний елемент «⊕» реалізує операцію заперечення еквіваленції (її ще називають сумою Жегалкіна, виключним додаванням, додаванням за модулем 2). Елементи «І-НЕ» і «АБО-НЕ» відповідно реалізують операції штрих Шеффера і стрілку Пірса.

Очевидно, що кількість базових елементів можна було б зменшити, залишивши лише ті, які відповідають якій-небудь одній функціонально повній системі логічних операцій. Теоретично основними базовими логічними операціями як правило називають заперечення, кон'юнкцію та диз'юнкцію і відповідно елементи «НЕ», «І», «АБО» - основними базовими логічними елементами . Проте на практиці з технологічних причин зручно буває мати дещо інший їх набір.

КС можна утворити, з'єднуючи виходи одних логічних елементів з входами інших. Схема, яка має n входів і один вихід, реалізує деяку n-місну логічну (булеву) функцію. Так, булевій функції, яка зображується формулою

$$f(x, y, z) = \bar{x} \wedge y \vee z$$

відповідатиме комбінаційна схема:



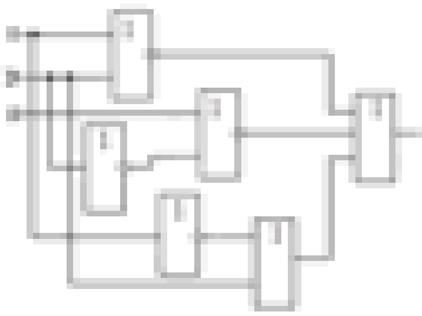
Часто доводиться розв'язувати задачі на аналіз та синтез схем.

Задача синтезу КС полягає в тому, щоб для заданої булевої функції побудувати комбінаційну схему, що її реалізує.

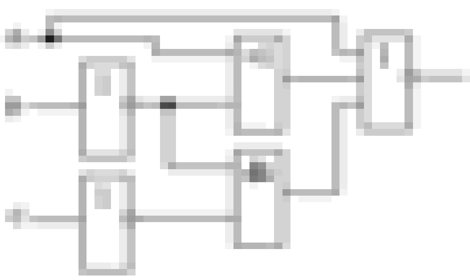
Задача аналізу комбінаційної схеми полягає в тому, щоб математично описати цю схему (побудувати її математичну модель), тобто знайти відповідну логічну формулу булевої функції. Таку задачу можна розв'язати, розкладаючи схему на підсхеми. Знайдену логічну формулу доцільно, по можливості спростити, замінивши такою рівносильною формулою, яка містить меншу кількість елементів.

Приклад 1. Реалізувати комбінаційною схемою наступну булеву функцію $f(x,y,z)=(x\downarrow y)\vee(\bar{y}\downarrow z)\vee(y\vee\bar{x})$.

Дана булева функція містить три змінні, які подаються на вхід нашої схеми, а також лише операції, що відповідають базовим логічним елементам комбінаційних схем: стрілка Пірса, заперечення, диз'юнкція. Порядок виконання операцій в булевій функції повинен відповідати послідовності з'єднань змінних логічними елементами та самих логічних елементів. Тому шукана комбінаційна схема має вид:



Приклад 2. Спростити комбінаційну схему:



Запишемо формулу булевої функції, яка відповідає даній схемі:

$f(a,b,c)=(a\oplus\bar{b})\downarrow a\downarrow(\bar{c}\wedge\bar{b})$

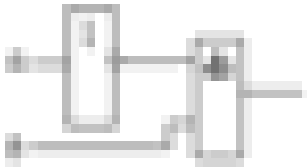
Нам потрібно отриману формулу максимально спростити:

$$(a\oplus\bar{b})\downarrow a\downarrow(\bar{c}\wedge\bar{b})=\overline{(\overline{a\leftrightarrow\bar{b}})\vee a\vee(\bar{c}\wedge\bar{b})}=(a\leftrightarrow\bar{b})\wedge\bar{a}\wedge(\overline{\bar{c}\wedge\bar{b}})=$$

$$=(a\to\bar{b})\wedge(\bar{b}\to a)\wedge\bar{a}\wedge(c\vee b)=(\bar{a}\vee\bar{b})\wedge(\bar{\bar{b}}\vee a)\wedge\bar{a}\wedge(c\vee b)=$$

$$=(b\vee a)\wedge\bar{a}\wedge(c\vee b)=[(b\wedge\bar{a})\vee(\overline{a\wedge}a)]\wedge(c\vee b)=b\wedge\bar{a}\wedge(c\vee b)=b\wedge\bar{a}$$

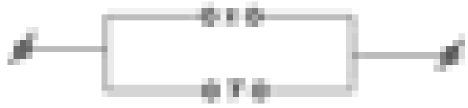
Для отриманої формули побудуємо комбінаційну схему:



Одним із способів фізичної реалізації комбінаційних схем є релейно-контактні схеми, які дають можливість реалізувати логічні операції шляхом замикання і розмикання контактів окремих електричних кіл.

Під *релейно-контактною* (РКС) схемою розуміють пристрій із провідників і двопозиційних контактів. Двопозиційний контакт – це фізичне тіло, яке може перебувати лише в двох станах – “ввімкнуто” і “вимкнуто”, які будемо позначати 1 і 0 відповідно. Контакти позначатимемо малими латинськими літерами.

Між собою контакти можуть з’єднуватись послідовно і паралельно:



Послідовне з’єднання провідників реалізує операцію кон’юнкції (сигнал через з’єднання контактів проходить тоді і тільки тоді, коли він проходить через кожний контакт).

Паралельне з’єднання провідників реалізує операцію диз’юнкції (сигнал через з’єднання контактів не проходить тоді і тільки тоді, коли він не проходить через жодний контакт з’єднання).

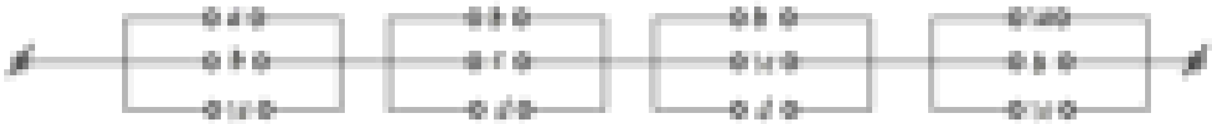
Послідовне з’єднання контактів позначатимемо xy, а їх паралельне з’єднання $x+y$. Через $\overline{x}$ будемо позначати контакт, який проводить сигнал тоді і тільки тоді, коли сигнал x його не проводить.

Отже, кожній РКС відповідає деяка формула алгебри висловлень, утворена за допомогою функціонально повної системи логічних операцій $\{\neg, \wedge, \vee\}$ і, навпаки, кожній такій формулі відповідає деяка контактна схема.

Аналогічно тому, як і в логіці висловлень не цікавляться змістом висловлень, а тільки тим, істинні вони чи хибні, так і при розгляді структурної формули цікавляться тільки «значенням замкнутості» схеми (чи схема проводить сигнал). Тому функцію, яка реалізується схемою, називають *функцією провідності* схеми. Очевидно, що схема буде замкнена тоді і тільки тоді, коли значення її функції провідності дорівнює 1. Тому дві контактні схеми називають *рівносильними*, якщо при одних і тих самих значеннях вхідних і вихідних контактів кожна з них проводить сигнал тоді і тільки тоді, коли його проводить інша. Інакше, дві схеми рівносильні тоді і тільки тоді, коли їх структурні формули рівносильні.

Задачею аналізу РК схеми називають задачу знаходження булевої функції (функції провідності), яку реалізує ця схема. Задача спрощення РК схем називається *задачею мінімізації*, а задача побудови релейно-контактних схем, що реалізують задані булеві функції є *задачею синтезу РК схем*.

Приклад 3. Максимально спростити РК- схему:

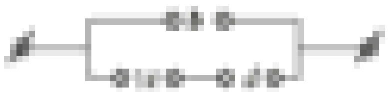


За схемою запишемо відповідну їй формулу контактних схем, спростимо формулу шляхом рівносильних перетворень і за спрощеною формулою побудуємо спрощену контактну схему:

$$(a+b+\overline{c})(b+c+d)(b+\overline{c}+d)(\overline{a}+b+\overline{c}) = b+(a+\overline{c})(c+d)(\overline{c}+d)(\overline{a}+\overline{c}) =$$

$$= b+[(a\overline{a}+\overline{c})(c\overline{c}+d)] = b+\overline{c}d$$

Отже, отримали спрощену схему, рівносильну заданій:



Запитання для самоконтролю

1. Що називають комбінаційною схемою (КС)?
2. З яких елементарних логічних елементів будують КС?
3. Яку булеву функцію реалізує кожен елементарний логічний елемент?
4. В чому суть задач синтезу та аналізу комбінаційної схеми?
5. Що називають релейно-контактною схемою (РКС)?
6. Які способи з’єднання контактів в РКС? Якими логічними операціями вони реалізуються?
7. Що називається функцією провідності РКС?
8. Які релейно-контактні схеми називають рівносильними?
9. В чому суть задач синтезу та аналізу РКС?
10. В чому суть задачі мінімізації релейно-контактної схеми?

ВПРАВИ

1.35 Побудувати комбінаційні схеми, що реалізують наступні булеві функції:

a) $y \oplus (x \downarrow (x \wedge z) \downarrow z)$;

b) $\overline{(x/y)\wedge(y \downarrow z)}$;

c) $(z \vee y) \oplus (y/(x \wedge z)/x)$;

d) $(z \to y) \oplus (y \leftarrow (x \wedge z))$;

e) $(x \leftarrow z \vee (z \wedge y)) \oplus (x \downarrow (x \wedge z) \downarrow z)$;

1.36 Побудувати РК-схеми, функції провідності яких визначаються наступними формулами:

a) $a(\bar{b}c + \bar{a} + b)$;

b) $\bar{a}(bc + \bar{b}\bar{c}) + a(\bar{b}c + b\bar{c})$;

c) $(a + \bar{b} + c)(\bar{a} + b)(c + \bar{d})(a + \bar{b} + d)$;

d) $\bar{a} + \bar{b} + \bar{c} + (\bar{a} + \bar{b} + c)(ab + \bar{c})$;

e) $(a + b)(\bar{b} + cd) + (\bar{a} + d + \bar{c})a\bar{b}$;

1.37 Максимально спростити РК-схеми:

a)

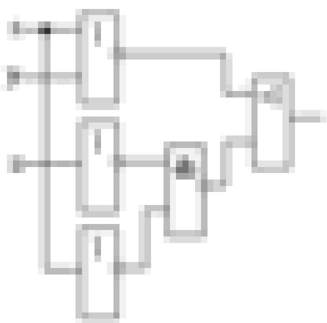
б)

в)

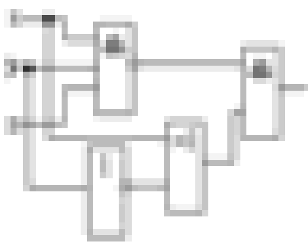
г)

1.38 Максимально спростити комбінаційні схеми.

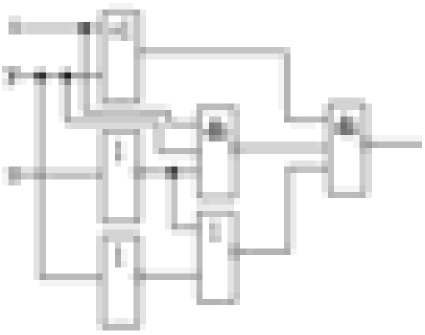
а)

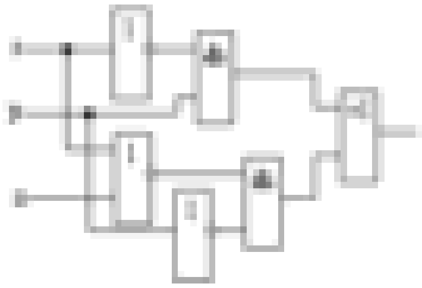


б)



в) г)





1.39 З n (n=2,3,4,…) контактів $x_1, x_2, \ldots, x_n$ скласти релейно-контактну схему, яка б спрацьовувала тоді й лише тоді, коли ввімкнено деякі, але не всі контакти.
1.40 Між поверхами двоповерхового будинку є одна лампа. Побудувати РК схему так, щоб на кожному поверсі можна було вимикачем включати і виключати світло незалежно від положення іншого вимикача.
1.41 Потрібно, щоб у великій залі можна було включати і виключати світло за допомогою будь-якого з чотирьох вимикачів, що розташовані на чотирьох стінах залу. Скласти відповідну РК схему.
1.42 Комітет складається з п’яти осіб, рішення приймається більшістю голосів. Якщо голова голосує проти, рішення не приймається. Скласти таку схему, щоб голосування відбувалось натисканням на кнопку, і у випадку, коли рішення прийняте, включалась лампа.
1.43 З n контактів $x_1, x_2, \ldots, x_n$ скласти релейно-контактну схему, яка б спрацьовувала тоді й тільки тоді, коли замкнено не більше *k* контактів.

1.6 ВІДНОШЕННЯ ЛОГІЧНОГО НАСЛІДКУ

Довести математичну теорему – це значить, виходячи з її умови, за певними правилами одержати висновок. У цьому випадку кажуть, що висновок теореми є «логічним наслідком» з умови або, що він виводиться з умови. Аналогічно проводяться будь-які доведення: будується ланцюжок тверджень, кожне з яких є або вихідним, або слідує з раніше встановлених, причому термін «слідує» не з’ясовується, а вважається відомим і цілком зрозумілим з логіки. Математична логіка дає точне визначення поняття слідування (вивідності).

Коли говорять, що з одного чи декількох тверджень $A_1, A_2, \ldots A_m$ слідує твердження *B*, то розуміють наступне: кожного разу, коли твердження $\mathbf{A_1}$, $A_1, A_2, \ldots A_m$ є істинними, істинним буде і твердження *B*.

Завдання алгебри висловлень полягає в тому, щоб вказати такі форми висловлень $A_1, A_2,...A_m$,*B*, при яких висловлення *B* обов'язково слідує з попередніх висловлень незалежно від конкретного змісту всіх цих висловлень. Але форми висловлень виражаються формулами алгебри висловлень. Тому теорія логічного слідування на базі алгебри висловлень вивчає закономірності утворення таких формул $A_1, A_2,...A_m$, *B*, що перші *m* формул зв'язані з останньою відношенням логічного слідування.

Означення. Формула *B логічно слідує* (є логічним наслідком) з формул $A_1, A_2,...A_m$, якщо вона набуває значення істинності «1» для кожного такого набору значень пропозиційних змінних, при якому всі формули $A_1, A_2,...A_m$ мають значення істинності «1».

Позначення: $A_1, A_2,...A_m \models B$.

При цьому формули $A_1, A_2,...A_m$ називають посилками, гіпотезами або припущеннями, а формулу *B* – логічним висновком, логічним наслідком.

Для порожньої множини посилок маємо $\models B$, що цілком узгоджується зі скороченням твердження «*B* є тавтологією».

Приклад 1. За таблицею істинності визначити, які з наступних формул: $a \to b$, $a \to (b \to c)$, $\bar{a} \vee (b \leftrightarrow c)$, є логічними наслідками інших.

Побудуємо таблиці істинності даних формул:

$\|a\|$	$\|в\|$	$\|c\|$	$\|a \to b\|$	$\|a \to (b \to c)\|$	$\|\bar{a} \vee (b \leftrightarrow c)\|$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	1

При з'ясуванні, чи є одна формула логічним наслідком множини формул, якщо побудовано їх таблиці істинності, треба визначити всі набори змінних, де досліджувана формула набуває значення 0. Якщо хоч на одному з цих наборів всі формули з множини формул набувають значення 1, то логічний наслідок не має місця.

Розглянемо формулу $a \to b$. З таблиці видно, що в 5-му рядку вона набуває значення 0, тому не є логічним наслідком формул $a \to (b \to c)$ та

$\overline{a} \vee (b \leftrightarrow c)$. Друга формула $a \to (b \to c)$ не є логічним наслідком формули $a \to b$, оскільки набуває значення 0 на 7-му наборі, де значення $/a \to b/=1$, але є логічним наслідком формули $\overline{a} \vee (b \leftrightarrow c)$. Очевидно, також, що третя формула $\overline{a} \vee (b \leftrightarrow c)$ є логічним наслідком з першої та другої формули: $a \to b$, $a \to (b \to c)$ $\models \overline{a} \vee (b \leftrightarrow c)$.

На практиці відношення логічного наслідку часто застосовується не до формул, а до висловлень, сформульованих природною мовою.

Означення. Нехай $\alpha_1, \alpha_2, ..., \alpha_m$ -міркування і β - висновок цих міркувань, а $A_1, A_2, ...A_m, B$ - формули алгебри висловлень, що відповідають їх логічним структурам. Кажуть, що міркування $\alpha_1, \alpha_2, ..., \alpha_m$ логічні і висловлення β є правильним (логічним) висновком, тоді і тільки тоді, коли формула *B* логічно слідує з формул $A_1, A_2, ...A_m$.

Приклад 2. Дослідити міркування на логічність методом від супротивного: «Якщо капіталовкладення залишаться постійними, то виростуть урядові витрати або виникне безробіття. Якщо урядові витрати не виростуть, то податки будуть знижені. Якщо податки будуть знижені і капіталовкладення залишаться постійними, то безробіття не виникне. Отже, урядові витрати виростуть.»

Розв’язання. Побудуємо логічну структуру цього міркування. Для цього спочатку виділимо прості висловлення:

«Капіталовкладення залишаться постійними» - *a*;

«Виростуть урядові витрати» - *b*;

«Виникне безробіття» - *c*;

«Податки будуть знижені» - d.

Логічною структурою міркування буде наступна послідовність формул:

$A_1 = a \to b \vee c,\ A_2 = \overline{b} \to d,\ A_3 = d \wedge a \to \overline{c},\ B = b$. Міркування буде логічним, за означенням, тоді і тільки тоді, коли формула *B* логічно слідує з формул A_1, A_2, A_3. Тобто $.\ a \to b \vee c, \overline{b} \to d, d \wedge a \to \overline{c} \models b$.

Припустимо, від супротивного, що логічний наслідок стоїть невірно, тобто існує набір змінних (a, b, c, d) для яких виконується система умов:

$$\begin{cases} |a \to b \vee c| = 1 \\ |\bar{b} \to d| = 1 \\ |d \wedge a \to \bar{c}| = 1 \\ |b| = 0 \end{cases} \Rightarrow \begin{cases} |a \to 0 \vee c| = 1 \\ |1 \to d| = 1 \\ |d \wedge a \to \bar{c}| = 1 \\ |b| = 0 \end{cases} \Rightarrow \begin{cases} |a \to c| = 1 \\ |d| = 1 \\ |a \to \bar{c}| = 1 \\ |b| = 0 \end{cases} \Rightarrow \begin{cases} |a| = 0 \\ |d| = 1 \\ |c| \in \{0,1\} \\ |b| = 0 \end{cases}.$$

Таким чином знайдено такі набори змінних ((0,0,0,1), (0,0,1,1)), при яких кожне з припущень буде істинним, а висновок – хибним. Отже, висновок, що урядові витрати виростуть не є правильним, а міркування не логічні.

Теорема 1.6.1. (критерій 1 логічного наслідку). Формула *B* логічно слідує з множини формул $A_1, A_2,...A_m$ тоді і тільки тоді, коли формула $A_1 \wedge A_2 \wedge ... \wedge A_m \to B$ є тавтологією.

Наслідки: 1) якщо серед посилок є хоч одна тотожно хибна, то будь-яка логічна формула *B* буде логічним наслідком з цих посилок;

2) якщо $\models B$, то $A_1, A_2,...A_m \models B$ для будь-яких $A_i, \left(i = \overline{1,m}\right)$.

Теорема 1.6.2. (критерій 2 логічного наслідку). Формула *B* логічно слідує з множини формул $A_1, A_2,...A_m$ тоді і тільки тоді, коли формула $A_m \to B$ логічно слідує з множини формул $A_1, A_2,...A_{m-1}$.

Наслідок. Формула *B* логічно слідує з множини формул $A_1, A_2,...A_m$ тоді і тільки тоді, коли формула $A_1 \to (A_2 \to ...(A_m \to B)...)$ є тавтологією.

Теорема 1.6.3 .Для логічного слідування мають місце такі властивості:

1. . $A_1, A_2,...A_m \models A_i$, $\left(i = \overline{1,m}\right)$ (властивість рефлексивності);
2. якщо $A \models B$ і $B \models C$ то $A \models C$ (властивість транзитивності);
3. якщо $A_1, A_2,...A_m \models B$ і A – довільна формула, то $A_1, A_2,...A_m, A \models B$ (приєднання довільної формули алгебри висловлень до числа посилок не порушує даного логічного слідування);
4. якщо $A_1, A_2,...A_m \models B$ і $\models A_i$, то $A_1,..., A_{i-1} A_{i+1},..., A_m \models B$ (вилучення з числа посилок формули, яка є тавтологією, не порушує логічного слідування).

Розглянемо основні *схеми логічного слідування,* які характеризують структуру правильного логічного мислення.

1. Правило МР (*modus ponens*): $\dfrac{A, A \to B}{B}$.

Тут і далі використовуємо нотацію, в якій над рискою записують посилки, під рискою – висновок, тобто цей запис означає $A, A \to B \models B$. Це правило має кілька назв: *правило висновку, правило відокремлення*.

2. MT (*modus tollens*): $\dfrac{A \to B, \overline{B}}{\overline{A}}$.

3. *Правило вилучення кон'юнкції*: $\dfrac{A \wedge B}{A}, \quad \dfrac{A \wedge B}{B}$.

4. *Правило введення кон'юнкції*: $\dfrac{A, B}{A \wedge B}$.

5. *Правило вилучення диз'юнкції*: $\dfrac{A \to C, B \to C}{A \vee B \to C}$.

6. *Правило введення диз'юнкції*: $\dfrac{A}{A \vee B}, \quad \dfrac{B}{A \vee B}$.

7. *Правило силогізму*: $\dfrac{A \to B, B \to C}{A \to C}$.

8. *Правило контрапозиції*: $\dfrac{A \to B,}{\overline{B} \to \overline{A}}$.

9. Узагальнення правила введення кон'юнкції: $\dfrac{A_1, A_2, \ldots, A_m}{A_{i_1} \wedge A_{i_2} \wedge \ldots \wedge A_{i_k}}, \ (i_j = \overline{1, m}, \quad j = \overline{1, k})$.

Для перевірки того, чи є дана формула *B* логічним наслідком із даних гіпотез $A_1, A_2, \ldots A_m$ - можна скористатися теоремами 1.6.1 та 1.6.2, або провести ланцюжок міркувань, застосовуючи схеми логічного слідування. Процес потрібно продовжувати до тих пір, поки не одержимо формулу *B*. Всі проміжні формули, які при цьому одержуються, будуть також логічними висновками гіпотез.

Приклад 3. Перевірити, чи має місце логічне слідування $A \wedge B, \ \overline{C} \to \overline{B} \models C$ і якщо так, то побудувати дедуктивний ланцюжок міркувань, який веде від посилок до висновку.

Пересвідчимося, що формула $(A \wedge B) \wedge (\overline{C} \to \overline{B}) \to C$ є тавтологією:

$(A \wedge B) \wedge (\overline{C} \to \overline{B}) \to C \equiv \overline{A \wedge B \wedge (C \vee \overline{B})} \vee C \equiv \overline{A} \vee \overline{B} \vee (\overline{C} \wedge B) \vee C \equiv$

$\equiv (\overline{A} \vee \overline{B} \vee \overline{C} \vee C) \wedge (\overline{A} \vee \overline{B} \vee B \vee C) \equiv 1$. Тому логічне слідування має місце.

Тепер побудуємо дедуктивний ланцюжок:

1. $A \wedge B$;
2. B (отримаємо з 1. за правилом вилучення кон'юнкції);
3. $\overline{C} \to \overline{B}$;
4. $\overline{\overline{C}}$ (з 2. і 3. за правилом МТ);
5. $\overline{\overline{C}} \to C$ (на основі закону $\overline{\overline{C}} \leftrightarrow C$ і правила вилучення кон'юнкції);
6. C (з 4. і 5. за правилом МР).

З логічним слідуванням пов'язане питання про сумісність формул алгебри висловлень.

Означення. Множина формул $\Gamma = \{A_1(x_1, x_2, ..., x_n), ..., A_m(x_1, x_2, ..., x_n)\}$ логіки висловлень називається *сумісною*, якщо існує хоча б один набір логічних змінних $x_1, x_2, ..., x_n$, при якому $|A_1 \wedge A_2 \wedge ... \wedge A_m| = 1$.

Множина формул, яка не є сумісною називається *суперечливою*.

Теорема 1.6.4. Якщо Γ – суперечлива множина формул алгебри висловлень, то для довільної формули *B* має місце слідування $\Gamma \models B$.

Приклад 4. Чи буде суперечливою множина формул: $a \to \overline{b}$, $\overline{a} \to \overline{c}$, $a \vee c$, $c \to b$?

Знайдемо для кожної формули множину наборів пропозиційних змінних, де вона набуває істинних значень, тобто її область істинності:

$$I_1 = \{(a,b,c) \in \{0,1\}^3 \setminus |a \to \overline{b}| = 1\} = \{0,1\}^3 \setminus \{(1,1,0),(1,1,1)\},$$

$$I_2 = \{(a,b,c) \in \{0,1\}^3 \setminus |\overline{a} \to \overline{c}| = 1\} = \{0,1\}^3 \setminus \{(0,0,1),(0,1,1)\},$$

$$I_3 = \{(a,b,c) \in \{0,1\}^3 \setminus |a \vee c| = 1\} = \{0,1\}^3 \setminus \{(0,1,0),(0,0,0)\},$$

$$I_4 = \{(a,b,c) \in \{0,1\}^3 \setminus |c \to b| = 1\} = \{0,1\}^3 \setminus \{(0,0,1),(1,0,1)\}.$$

Оскільки, $I_1 \cap I_2 \cap I_3 \cap I_4 = \{(1,0,0)\} \neq \varnothing$, то дана множина формул не є суперечливою.

Можна задачу поставити іншим чином, наприклад: знайти всі формули *B*, які є логічними висновками з даних гіпотез $A_1, A_2, ... A_m$.

Всяка тотожно істинна формула (тавтологія) є логічним висновком з будь-якої системи гіпотез (бо завжди $|A \to 1| = 1$) і логічними висновками з тотожно істинних гіпотез є будь-які тотожно істинні формули, і тільки вони (бо $|1 \to B| = 1$ тоді і тільки тоді, коли $|B| = 1$). Тоді, за умов, що не всі гіпотези $A_1, A_2, ... A_m$ є тавтологіями, і серед логічних висновків з них не враховуватимемо тавтологій, справедлива наступна теорема.

Теорема 1.6.5. Формула *B* є логічним наслідком з гіпотез $A_1, A_2, ... A_m$ тоді і тільки тоді, коли, коли елементарні диз'юнкції ДКНФ(В) містяться серед елементарних диз'юнкцій ДКНФ ($A_1 \wedge A_2 \wedge ... \wedge A_m$).

Існують комп'ютерні програми, які розроблено для автоматизації міркувань, що виконуються за допомогою доведення логічних теорем. У багатьох із цих програм використано правило виведення, відоме як резолюція. Правило резолюції було запропоноване у 1930 році Жаком Ербраном для доведення теорем у формальних системах першого порядку. На основі правила резолюції Джон Алан Робінсон у 1965 році запропонував *метод резолюцій* автоматичного доведення логічних теорем, який базується на доведенні від супротивного.

Нехай потрібно довести, що істинна деяка формула А. Тоді розглядаємо заперечення формули А, і зводимо $\overline{A}$ до кон'юнктивної нормальної форми (КНФ). Нагадаємо, що КНФ – це формула, що рівносильна даній формулі і є кон'юнкцією диз'юнктивних одночленів $\overline{A} \equiv D_1 \wedge ... \wedge D_n$, де D_i - є диз'юнкція скінченного числа пропозиційних змінних або їй заперечень (літералів). Тим самим формується множина диз'юнктів $K = \{D_1, ..., D_n\}$ - клаузальна множина. Два диз'юнкти, що містять пропозиційні змінні з протилежними знаками, - контрарні літерали, наприклад, $D_i = D_i^{/} \vee x$, $D_k = D_k^{/} \vee \overline{x}$, формують третій диз'юнкт – резольвенту $D_i^{/} \vee D_k^{/}$, з якої виключено контрарні літерали, Тобто має місце правило логічного виводу, що носить назву правила резолюції:

$$\frac{D_i^{/} \vee x, \quad D_k^{/} \vee \overline{x}}{D_i^{/} \vee D_k^{/}}.$$

Для його обґрунтування, досить переконатися, що має місце тавтологія $\models \left(D_i^{/} \vee x\right) \wedge \left(D_k^{/} \vee \overline{x}\right) \to D_i^{/} \vee D_k^{/}$. Дійсно, виконуючи ланцюг рівносильних перетворень, одержимо:

$$\left(D_i^{/} \vee x\right) \wedge \left(D_k^{/} \vee \overline{x}\right) \to D_i^{/} \vee D_k^{/} \equiv \overline{\left(D_i^{/} \vee x\right)} \vee \overline{\left(D_k^{/} \vee \overline{x}\right)} \vee D_i^{/} \vee D_k^{/} \equiv$$

$$\equiv \left(\overline{D_i^{/}} \wedge \overline{x}\right) \vee \left(\overline{D_k^{/}} \wedge x\right) \vee D_i^{/} \vee D_k^{/} \equiv \left(\overline{D_i^{/}} \wedge \overline{x}\right) \vee D_i^{/} \vee \left(\overline{D_k^{/}} \wedge x\right) \vee D_k^{/} \equiv$$

$$\equiv \left(\overline{D_i^{/}} \vee D_i^{/}\right) \wedge \left(\overline{x} \vee D_i^{/}\right) \vee \left(\overline{D_k^{/}} \vee D_k^{/}\right) \wedge \left(x \vee D_k^{/}\right) \equiv \overline{x} \vee D_i^{/} \vee x \vee D_k^{/} \equiv 1 \vee D_i^{/} \vee D_k^{/} \equiv 1.$$

Поступово застосовуючи правило резолюції до множини диз'юнктів прагнемо отримати порожній диз'юнкт. Поява порожнього диз'юнкта вказує на протиріччя, оскільки порожня резольвента одержується з двох суперечних диз'юнктів x та $\overline{x}$, кожен з яких є логічним наслідком формули $\overline{A}$. Це завершує доведення висновком, що $\overline{A}$ хибне, отже А істинне.

Дж. Робінсон показав, що метод резолюції є коректним і повним в тому розумінні, що порожній диз'юнкт виводиться із вхідної множини диз'юнктів шляхом деякої послідовності кроків резолюції тоді і тільки тоді, коли множина диз'юнктів суперечлива.

Саме по собі правило резолюції утворює повну систему правил виводу для множини диз'юнктів логіки висловлень. Іншими словами, при використанні одного правила резолюції можна вивести будь-який наслідок з множини формул, які подані у вигляді множини диз'юнктів.

Приклад 5. Довести логічний наслідок методом резолюцій:

$\overline{P} \vee \overline{Q} \vee R, \overline{P} \vee \overline{Q} \vee S \models \overline{P} \vee S \vee \overline{Q}$.

Посилки уже записані у кон`юнктивній нормальній формі. Отже, їм відповідатиме така множина диз'юнктів: $\{\overline{P} \vee \overline{Q} \vee R\}$, $\{\overline{P} \vee \overline{Q} \vee S\}$.

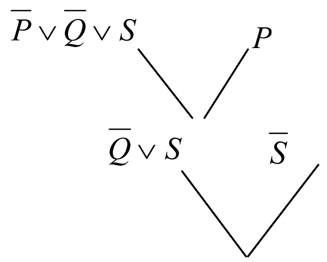
Подамо заперечення висновку у кон`юнктивній нормальній формі $\overline{(\overline{P} \vee S \vee \overline{Q})} \equiv \overline{\overline{P}} \wedge \overline{S} \wedge \overline{\overline{Q}} \equiv P \wedge \overline{S} \wedge Q$.

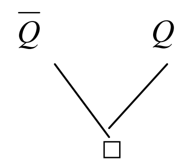
Отже, отримали таку клаузальну множину: $\{\overline{P} \vee \overline{Q} \vee R, \overline{P} \vee \overline{Q} \vee S, P, \overline{S}, Q\}$.

Для окремої зручності запишемо елементи (диз'юнкти) цієї множини в позначеннях: $\alpha_1 = \overline{P} \vee \overline{Q} \vee R$; $\alpha_2 = \overline{P} \vee \overline{Q} \vee R$; $\alpha_3 = P$; $\alpha_4 = \overline{S}$; $\alpha_5 = Q$.

До диз'юнктів $\alpha_1 - \alpha_5$ застосуємо резолюційний процес. Застосувавши правило резолюцій до диз'юнктів α_3 і α_2, дістанемо резольвенту $\alpha_6 = \overline{Q} \vee S$. α_6 і α_4 породжують резольвенту $\alpha_7 = \overline{Q}$, а α_7 і α_5 породжують порожній диз'юнкт □. Це означає, що співвідношення має місце.

Алгоритм резолюційного процесу зручно подати у вигляді дерева. Так, для наведеного прикладу дедуктивне дерево матиме такий вигляд:





Зазначимо, що оскільки доведення логічного слідування $\overline{P} \vee S \vee \overline{Q}$ з припущень $\overline{P} \vee \overline{Q} \vee R$ і $\overline{P} \vee \overline{Q} \vee S$ рівносильне дослідженню формули $\left(\overline{P} \vee \overline{Q} \vee R\right) \wedge \left(\overline{P} \vee \overline{Q} \vee S\right) \rightarrow \overline{P} \vee S \vee \overline{Q}$ на тотожну істинність, то задачу можна розв'язувати й як перевірку на невиконуваність заперечення цієї формули. Подамо заперечення її в кон`юнктивній нормальній формі: $\left(\overline{P} \vee \overline{Q} \vee R\right) \wedge \left(\overline{P} \vee \overline{Q} \vee S\right) \wedge P \wedge \overline{S} \wedge Q.$ Отже, маємо такий самий, як і в попередньому випадку, набір диз'юнктів. Застосувавши до них резолюційний процес, отримуємо порожній диз'юнкт. А це означає, що заперечення формули є суперечністю. Отже, сама формула є тавтологією.

<u>Запитання для самоконтролю</u>

1. Яка формула *B* логічно слідує (називається логічним наслідком) з формул $A_1, A_2, \ldots A_m$?
2. Як узгоджується позначення логічного наслідку та тавтології?
3. Які міркування називають логічними, а висновок правильним?
4. Сформулюйте критерії 1 та 2 логічного наслідку.
5. Які властивості має відношення логічного наслідку?
6. Як формулюються наступні схеми логічного слідування: 1) modus ponens, 2) modus tollens, 3) правило вилучення кон'юнкції, 4) правило контра позиції, 5) правило вилучення диз'юнкції, 6) правило введення диз'юнкції, 7) правило силогізму, 8) правило введення кон'юнкції?
7. Яка множина формул алгебри висловлень називається сумісною (суперечливою)?
8. Як шукати логічні наслідки з гіпотез $A_1, A_2, \ldots A_m$, що не всі є тавтологіями?
9. Яке правило логічного слідування носить назву правила резолюції?
10. В чому суть методу резолюцій? В яких задачах можна його використовувати?

<u>ВПРАВИ</u>

1.44 Чи правильно стоїть знак логічного наслідку у співвідношеннях:

a) $(a \wedge b) \rightarrow c, (a \vee b) \rightarrow \overline{c} \ \models a \wedge b \wedge c$;

b) $a \wedge b \wedge \overline{c}, b \rightarrow c, b \rightarrow a \models b \rightarrow (a \vee c)$;

c) $a \to b, a \to \overline{b} \models \overline{a}$;

d) $a \to b, c \to d, \overline{b} \to \overline{d} \models \overline{a} \to \overline{b}$;

e) $(a \wedge b) \to c, \overline{d} \to (e \to f), c \to (e \to f) \models a \to (b \to d)$.

1.45. Довести, що для довільних формул *A, B, C* алгебри висловлень мають місце співвідношення логічного наслідку:

a) MT (modus tollens): $\dfrac{A \to B, \overline{B}}{\overline{A}}$;

b) Правило вилучення кон'юнкції: $\dfrac{A \wedge B}{A}$, $\dfrac{A \wedge B}{B}$;

c) Правило введення кон'юнкції: $\dfrac{A, B}{A \wedge B}$;

d) Правило вилучення диз'юнкції: $\dfrac{A \to C, B \to C}{A \vee B \to C}$;

e) Правило введення диз'юнкції: $\dfrac{A}{A \vee B}$, $\dfrac{B}{A \vee B}$;

1.46. Пересвідчитись, що для довільних формул *A, B, C* алгебри висловлень справедливі твердження:

a) з $A \models B$ та $B \models C$ випливає $A \models C$;

b) з $A, B \models C$ випливає $A \models B \to C$;

c) з $A \models B$ випливає $\overline{B} \models \overline{A}$;

d) з $A \to B \models C$ випливає $A \wedge B \models C$;

e) якщо $A \vee B$ тавтологія, то $\overline{A} \models B$ і $\overline{B} \models A$;

f) якщо $A \vee \overline{B}$ тавтологія, то і $B \models A$;

1.47 Користуючись поняттям логічного наслідку з'ясувати, чи є логічними наступні міркування:

a) Якщо Іван є учасником олімпіади з математики, то він обов'язково розумний і добре володіє матеріалом. Але він не учасник цієї олімпіади. Отже, він або не розумний, або ж не володіє матеріалом.

b) Студент не складе екзамен, якщо погано підготується до нього або захворіє. Якщо ж він не складе екзамен, то не буде отримувати стипендію. Але студент не хворий. Отже, якщо студент погано підготується до екзамену, то він не матиме стипендії.

c) Якщо число розкладається на добуток s різних простих чисел, то воно має 2^s різних дільників. Дане число має точно 2^s різних дільників. Отже, воно розкладається на добуток s різних простих чисел;

d) Якщо заданий чотирикутник – ромб, то його діагоналі взаємно перпендикулярні. Якщо діагоналі чотирикутника не взаємно перпендикулярні, то він не є квадратом. Якщо даний чотирикутник – квадрат, то його можна вписати в коло. Неправильно, що даний чотирикутник не можна вписати в коло, або його діагоналі взаємно перпендикулярні. Отже, даний чотирикутник не є ні ромбом, ні квадратом;

e) Якщо Іван поїде в Київ, то Сергій поїде в Одесу. Іван поїде в Київ або в Донецьк. Якщо Іван поїде в Донецьк, то Роман залишиться дома. Але Роман дома не залишиться, отже, Іван поїде в Київ.

1.48 Перевірити, чи буде суперечливою множина формул:

a) $(a \vee b) \to c, \overline{a} \wedge b \wedge c$;

b) $(\overline{a} \leftrightarrow b) \vee c, \overline{a} \wedge b \wedge \overline{c}, a \wedge \overline{b} \wedge \overline{c}$;

c) $\overline{a} \to \overline{b}, c \to b, c \wedge \overline{a}$;

d) $a \leftrightarrow \overline{b}, c \to b, \overline{a} \to \overline{c}, a \vee c$.

1.49 Перевірити, чи є сумісною множина висловлень:

a) Якщо контрольна робота буде на наступному тижні, то Наташа підготується до неї. Вона не підготується до контрольної роботи, якщо поїде на змагання. Наташа поїде на змагання тоді тільки тоді, коли одержить стипендію. Контрольна робота буде на наступному тижні і Наташа не одержить до того часу стипендії.

b) Якщо водій автобуса порушив правила руху, то свідок говорить правду. Якщо в момент аварії на світлофорі було зелене світло, то свідок говорить неправду. Мотоцикліст порушив правила руху тоді і тільки тоді, коли на світлофорі в момент аварії було зелене світло. Як водій автобуса,так і мотоцикліст порушили правила руху;

c) Якщо курс цінних паперів росте або зменшується процентна ставка, то або падає курс акцій, або податки не збільшуються. Курс акцій падає тоді і тільки тоді, коли росте курс цінних паперів і збільшуються податки. Якщо зменшується процентна ставка, то або курс акцій не падає, або курс цінних паперів не росте. Або збільшуються податки, або падає курс акцій і зменшується процентна ставка.

1.50 Знайти всі нерівносильні між собою і не тотожно істинні формули алгебри висловлень, які є логічними наслідками множини формул:

a) $\overline{a} \to \overline{b}, c \to b, \overline{c} \wedge a$;

b) $a \leftrightarrow \overline{b}, c \leftrightarrow b, a \to \overline{c}, a \vee c$;

c) $\left(\overline{a} \to b\right) \to \overline{a}, a \to c, a \wedge b \to c$;

d) $\left(a \to c\right) \wedge \overline{a}, b \to \overline{c}, \overline{a} \vee \overline{b}$.

1.51 Довести логічний наслідок методом резолюцій для схем із завдання 1.45.

1.52 Користуючись методом резолюцій перевірити, чи є формула тавтологією:

a) $\left(a \vee b\right) \to (c \to a \vee b)$;

b) $\left(a \to \left(b \to c\right)\right) \to \left(\left(a \to b\right) \to \left(a \to c\right)\right)$;

c) $\left(a \to b\right) \to \left(\left(a \to c\right) \to \left(a \to b \wedge c\right)\right)$;

d) $\left(a \to c\right) \to \left(\left(b \to c\right) \to \left(a \vee b \to c\right)\right)$;

e) $\left(a \to b\right) \to \left(\overline{b} \to \overline{a}\right)$.

2. ЧИСЛЕННЯ ВИСЛОВЛЕНЬ

В основу викладу алгебри висловлень було покладено поняття висловлення. При цьому вимагалось, щоб будь-яке висловлення було або істинним, або хибним. Ця вимога (закон виключеного третього) стосувалась нескінченної сукупності всіх висловлень. Такий підхід не прийнятний в основах математики, зокрема там, де необхідно обґрунтувати, що використання цього закону не приводить до суперечності.

Розглянемо формалізовану аксіоматичну теорію, яка адекватна алгебрі висловлень і вільна від указаної вище вимоги. Цю формалізовану аксіоматичну теорію прийнято називати *численням висловлень.* Числення висловлень є складовою частиною інших логічних числень.

Формалізація будь-якої змістовної теорії передбачає перетворення її на об'єкт вивчення. Для цього спочатку чітко описується мова даної теорії, а саме:

- вказується алфавіт теорії - множина усіх її вихідних символів (Символи теорії розглядаються як матеріальні знаки, які означають лише те, що про них буде сказано в аксіомах, і з якими працюють тільки так, як це буде сказано в правилах перетворень (виводів). При цьому символи, що не належать до алфавіту теорії, називаються *метасимволами);*
- серед слів (скінченних послідовностей символів), записаних в алфавіті даної теорії, виділяють ті, що називаються *формулами;*
- з класу формул виділяються *аксіоми*;
- вказуються точно правила переходу від одних формул даної теорії до інших формул цієї ж теорії. їх називають *правилами виводу*.

Аксіоми і правила виводу вибирають так, щоб з множити аксіом за допомогою правил виводу можна одержати всі тотожно істинні формули змістовної теорії, і тільки їх. Таким чином, тотожно істинні формули змістовної теорії виявляються теоремами відповідної формальної теорії.

При побудові числення висловлень можуть бути вибрані різні системи аксіом і вказані різні правила виводу. Проте спільним для них є те, що кожного разу множина формул числення висловлень, яку можна одержати за допомогою вказаних правил виводу з заданої системи аксіом, збігається з множиною тавтологій алгебри висловлень.

2.1 АЛФАВІТ, ФОРМУЛИ, АКСІОМИ, ПРАВИЛА ВИВОДУ ЧИСЛЕННЯ ВИСЛОВЛЕНЬ L_1. ПРИКЛАДИ ДОВЕДЕНЬ В ЧИСЛЕННІ L_1

Розглянемо одну з можливих аксіоматизацій алгебри висловлень, яку коротко називатимемо численням висловлень L_1.

1. *Алфавіт теорії L_1* складають:

1) символи першої категорії - малі латинські букви і ці ж букви з натуральними індексами (їх будемо називати *пропозиційними буквами*, або *пропозиційними змінними*);

2) символи другої категорії - логічні зв'язки: $\neg, \wedge, \vee, \rightarrow$. За ними зберігаються лише відповідні назви з алгебри висловлень: заперечення, кон'юнкція, диз'юнкція, імплікація;

3) символи третьої категорії – допоміжні або розділові: (,), (їх називають відповідно лівою і правою дужками);

4) інших символів в алфавіті теорії L_1, крім вказаних в пунктах 1) - 3), немає.

Скінченні послідовності символів алфавіту утворюють *слова*. Серед слів вибираються *формули.* Точне означення формули носить рекурсивний характер.

2. Формулами теорії L_1 є такі слова:

1) будь-яка пропозиційна буква - формула;

2) якщо A і B формули, то формулами будуть слова $(\neg A), (\neg B), (A \wedge B), (A \vee B), (A \rightarrow B)$;

3) інших формул теорії L_1, крім зазначених в пунктах 1) і 2), немає.

Звертаємо увагу на те, що символи для позначення формул – A, B, C, …. не є символами алфавіту теорії L_1, тобто це *метасимволи.* Вони використовуються для скороченого позначення формул.

Якщо формула є пропозиційною буквою, то її називають *елементарною формулою.*

Використання дужок для запису формул дозволяє поділити побудову формули на етапи і на кожному етапі перевіряти, чи є те чи інше підслово формулою. Оскільки таких кроків скінченне число, то питання про те, чи є дане слово формулою, завжди може бути вирішеним.

Як і в алгебрі висловлень, домовляються не записувати зовнішніх дужок. Крім цього, опускають дужки, враховуючи силу логічних зв'язок (за спаданням) $\neg, \wedge, \vee, \rightarrow$.

3. - Аксіомами числення висловлень L_1 є формули:

1.1 $a \to (b \to a)$;

1.2 $(a \to (b \to c)) \to ((a \to b) \to (a \to c))$;

2.1 $a \wedge b \to a$;

2.2 $a \wedge b \to b$;

2.3 $(a \to b) \to ((a \to c) \to (a \to b \wedge c))$;

3.1 $a \to a \vee b$;

3.2 $b \to a \vee b$;

3.3 $(a \to c) \to ((b \to c) \to (a \vee b \to c))$;

4.1 $(a \to b) \to (\overline{b} \to \overline{a})$;

4.2 $a \to \overline{\overline{a}}$;

4.3 $\overline{\overline{a}} \to a$.

Аксіоми розбиті на 4 групи залежно від наявності в них тих чи інших логічних зв'язок: 1 група - "$\to$"; 2 група - "$\wedge$", "$\to$"; 3 група - "$\vee$", "$\to$"; 4 група - "$\neg$", "$\to$".

4. Правила виводу в численні висловлень L_1:

1) *Правило підстановки.*

Нехай A - формула, яка містить пропозиційну букву a. Тоді, якщо A - вивідна формула числення L_1, то замінюючи в ній всі входження букви a довільною формулою B, одержимо вивідну формулу. Скорочений запис підстановки - $S_a^B(A)$

2) *Правило modus ponens* (скорочено MP).

Якщо A і $A \to B$ вивідні формули числення L1, то вивідною буде і формула B. Це правило називають також *правилом висновку* або *правилом відокремлення* і скорочено записують так: $\dfrac{A,\, A \to B}{B}$.

В формулюваннях правил виводу фігурує поняття "вивідна формула".

Означення. Формула A називається *вивідною в численні висловлень* L_1, якщо існує така скінченна послідовність формул $B_1, B_2, .., B_n$, в якій $B_n = A$ і кожна формула B_i $(i = \overline{1, n})$ є або аксіомою, або одержана з попередньої за допомогою правила підстановки, або одержана з двох попередніх за допомогою правила MP.

Вивідну формулу A називають *теоремою* числення висловлень L1 і записують $\vdash A$, а послідовність формул $B_1, B_2, .., B_n$ називається виводом (доведенням) формули A в численні L1. Число n називається довжиною виводу формули A..

Таким чином, кожна аксіома є вивідною формулою з довжиною виводу 1. Користуючись правилами виводу і виходячи з аксіом, одержують нові вивідні формули числення висловлень.

Приклад 1. Візьмемо аксіому 1.1 і виконаємо підстановку $S_a^{a\wedge c}$. Одержимо $\vdash a\wedge c \to (b \to a\wedge c)$ - теорема з довжиною виводу n=2.

В свою чергу, якщо в одержаній формулі здійснимо підстановки $S_a^{\overline{a\wedge c}}\ {}_b^{\bar{c}}$, то одержимо нову вивідну формулу (теорему) числення висловлень:

$\vdash (\overline{a\wedge c})\wedge c \to (\bar{c} \to (\overline{a\wedge c})\wedge c)$, яка має довжину виводу n=3.

Приклад 2. Довести, що формула $(a \to b)\to(a \to a)$ є теоремою, та визначити довжину побудованого виводу.

<u>Розв'язання.</u> Побудуємо вивід (доведення) для вказаної формули. Беремо аксіому 1.2 і здійснюємо підстановку S_c^a.

Одержимо: $\vdash (a \to (b \to a))\to((a \to b)\to(a \to a))$.

Оскільки $\vdash a \to (b \to a)$, бо це аксіома 1.1, то за правилом МР маємо $\vdash (a \to b)\to(a \to a)$.

Випишемо всі формули, що утворюють вивід:

1) $(a \to (b \to c))\to((a \to b)\to(a \to c))$- аксіома 1.2;

2) $(a \to (b \to a))\to((a \to b)\to(a \to a))$ - $S_c^a(1)$;

3) $a \to (b \to a)$ - аксіома 1.1;

4) $(a \to b)\to(a \to a)$ - правило МР(2,3).

Отже, довжина побудованого виводу дорівнює 4.

Приклад 3. Довести, що коли $\vdash I$, то $\vdash A \to I$ для будь-якої формули A.

<u>Розв'язання</u>. Візьмемо аксіому 1.1 і здійснимо підстановки $S_a^I\ {}_b^A$. Одержимо $\vdash I \to (A \to I)$. За умовою $\vdash I$. Тоді за правилом МР маємо $\vdash A \to I$.

Приклад 4. Довести, що $\vdash a \to a$.

Розв'язання. За доведеним в прикладі 2 маємо $\vdash (a \to b) \to (a \to a)$. В цій формулі зробимо підстановку $S_b^{b \to a}$. Одержимо: $\vdash (a \to (b \to a)) \to (a \to a)$. Оскільки $\vdash a \to (b \to a)$, бо це аксіома 1.1, то за правилом МР маємо $\vdash a \to a$, що і треба було довести.

Запитання для самоконтролю

1. Яку аксіоматичну теорію прийнято називати численням висловлень?
2. Які кроки опису мови довільної формальної теорії?
3. За яким принципом обирають аксіоми і правила виводу формальної теорії?
4. Які символи містить числення висловлень L_1?
5. Що називається формулою числення висловлень L_1?
6. Які групи аксіом має числення висловлень L_1 та що їх характеризує?
7. В чому суть правила підстановки ($S_a^B(A)$)?
8. В чому суть правила виведення modus ponens (MP)?
9. Яка формула називається вивідною в численні висловлень L_1? Що таке вивід, довжина виводу?
10. Що таке теорема числення висловлень L_1? Наведіть приклади теорем числення висловлень L_1.

2.2. ВИВІДНІСТЬ ІЗ ГІПОТЕЗ. МЕТАТЕОРЕМА ДЕДУКЦІЇ. ДОДАТКОВІ ПРАВИЛА ВИВОДУ.

Застосування тільки двох правил формального доведення значно ускладнює процес побудови виводів формул. Крім того, такі доведення відрізняються від багатьох змістовних доведень в математиці, бо в останніх, крім аксіом і доведених раніше теорем, широко використовують певні припущення. Тому для скорочення виводів і наближення їх до практики математичних доведень розглядають виводи з припущень (гіпотез) і вводять додаткові правила виводу.

Нехай Г - деяка множина формул числення висловлень.

Означення. Формула *А* називається *вивідною з множини формул Г*, якщо існує скінченна послідовність формул $B_1, B_2, .., B_n$, в якій $B_n = A$, і кожна формула є або формулою множини Г, або вивідною в численні висловлень L1 або одержана з попередніх за допомогою правила МР.

При цьому послідовність формул $B_1, B_2, .., B_n$ називається *виводом (доведенням) формули А з множини формул Г*. Формули з Г називаються *припущеннями* або *гіпо*тезами. Запис Г $|-A$ означає, що формула A вивідна з множини формул Г. В тому разі, коли Г=∅, маємо $|-A$, тобто A - теорема числення висловлень.

При розгляді формалізованих теорій розрізняють теореми формалізованої теорії (вивідні формули) і теореми про формалізовану теорію. Теореми числення висловлень в сукупності складають формалізовану теорію числення висловлень. А міркування про теорію, про окремі теореми цієї теорії, про зв'язки між теоремами і т.д. становлять так звану *метатеорію* даної формалізованої теорії - в нашому випадку метатеорію числення висловлень. Твердження метатеорії називатимемо *метатеоремами*. Доведення метатеорем не формалізовані, а змістовні.

Вкажемо деякі важливі властивості поняття вивідності з припущень.

Метатеорема 2.2.1 1) Якщо Г $|-A$ і Г ⊂Δ, Δ $|-A$.

2) Г $|-A$ тоді і тільки тоді, коли в Г є така скінченна або порожня підмножина Γ_1 , що Γ_1 $|-A$.

3) Якщо Г $|-A$ для будь-якої формули $A\in\Delta$ і $\Delta|-B$, то $\Gamma|-B$.

Метатеорема 2.2.2. (дедукції). Якщо $C_1,C_2,...C_{m-1},C_m|-A$, то $C_1,C_2,...C_{m-1}|-C_m\to A$.

Наслідок 1. $C_1,C_2,...C_{m-1},C_m|-A$ тоді і тільки тоді, коли $C_1,C_2,...C_{m-1}|-C_m\to A$.

Наслідок 2. $C_1,C_2,...C_{m-1},C_m|-A$ тоді і тільки тоді, коли $|-C_1\to(C_2\to...\to(C_{m-1}\to(C_m\to A))...)$.

Другий наслідок ілюструє можливість переходу від вивідності з припущень до теорем числення висловлень, і навпаки.

Приклад 5. Довести, що $|-(a\to b)\to((b\to c)\to(a\to c))$.

Розв'язання. Використовуючи наслідок 2 з метатеореми дедукції, маємо: $|-(a\to b)\to((b\to c)\to(a\to c))\Leftrightarrow\quad(a\to b),(b\to c),a|-c$. Побудуємо виведення з припущень:

1. $(a\to b)$ припущення;
2. $(b\to c)$ припущення;

3.	a	припущення;
4.	b	MP(1,3);
5.	c	MP(2,4).

Таким чином, $(a \to b),(b \to c), a \,|\!- c$. Отже $|\!- (a \to b) \to ((b \to c) \to (a \to c))$.

Тепер нехай *A,B,C* – довільні формули числення висловлень. Виконаємо підстановку $S_{a\ b\ c}^{A\ B\ C}$ в одержану вивідну формулу:

$$|\!- (A \to B) \to ((B \to C) \to (A \to C)).$$

Отже, якщо $|\!- (A \to B)$ і $|\!- (B \to C)$, то за правилом MP буде і $|\!- (A \to C)$. Таким чином ми одержали нове правило виводу, яке називається *правилом силогізму:* $\dfrac{A \to B, B \to C}{A \to C}$.

Метатеорема 2.2.3 В численні висловлень мають місце наступні *похідні правила виведення*:

1. *Правило силогізму*: $\dfrac{A \to B, B \to C}{A \to C}$.
2. *Правило вилучення кон'юнкції*: $\dfrac{A \wedge B}{A}$, $\dfrac{A \wedge B}{B}$.
3. *Правило введення кон'юнкції*: $\dfrac{A, B}{A \wedge B}$.
4. *Правило вилучення диз'юнкції*: $\dfrac{A \to C, B \to C}{A \vee B \to C}$.
5. *Правило введення диз'юнкції*: $\dfrac{A}{A \vee B}$, $\dfrac{B}{A \vee B}$.
6. MT (*modus tollens*): $\dfrac{A \to B, \overline{B}}{\overline{A}}$.
7. *Правило контрапозиції*: $\dfrac{A \to B,}{\overline{B} \to \overline{A}}$.

Приклад 6. Провести змістовне і формальне доведення теореми про три перпендикуляри: «Для того щоб пряма, що лежить у площині, була перпендикулярна до похилої, необхідно і достатньо, щоб ця пряма була перпендикулярна до проекції похилої».

<u>Розв'язання</u>. Розглянемо доведення достатності (доведення необхідності аналогічне і рекомендується для самостійного опрацювання).

Змістовне доведення. Позначимо через AB – похилу, а *CB* – проекцію похилої AB на площину α. Нехай пряма m, що лежить у площині α, перпендикулярна до CB. Із означення перпендикуляра до площини (AC⊥α) випливає, що AC⊥m. Отже, пряма m перпендикулярна до прямих CB і AC, що лежать у площині (ABC). Тоді m⊥(ABC). Звідки m⊥AB.

Формальне доведення.		Аналіз доведення
1.	$CB=Пр_{\alpha}AB$;	припущення;
2.	$(CB=Пр_{\alpha}AB)\to(AC\perp\alpha)$;	з означення проекції;
3.	$(AC\perp\alpha)$;	MP(1,2);
4.	$m\subset\alpha$;	припущення;
5.	$(AC\perp\alpha)\wedge(m\subset\alpha)$;	правило введення $\wedge$ (ВК);
6.	$(AC\perp\alpha)\wedge(m\subset\alpha)\to(AC\perp m)$;	з означення перпендикулярності прямої та площини;
7.	$AC\perp m$;	MP(5,6);
8.	$m\perp BC$;	припущення;
9.	$(m\perp AC)\wedge(m\perp BC)$;	ВК(7,8);
10.	$(m\perp AC)\wedge(m\perp BC)\to(m\perp(ABC))$	раніше доведена теорема;
11.	$m\perp(ABC)$;	MP(9,10);
12.	$AB\subset(ABC)$;	аксіома;
13.	$(m\perp(ABC))\wedge(AB\subset(ABC))$;	ВК(11,12);
14.	$(m\perp(ABC))\wedge(AB\subset(ABC))\to(m\perp AB)$	з означення;
15.	$m\perp AB$	MP(13,14)

Запитання для самоконтролю

1. Яка формула називається *A* називається вивідною з множини формул Г? Що називається припущеннями або гіпотезами у цій вивідності?
2. Що називається виводом (формальним доведенням) з припущень та чому його відмінності з формальним виводом теорем числення висловлень L_1?
3. Що таке метатеорія даної формалізованої теорії? Як називаються її твердження?
4. Які властивості має вивідність з припущень у числення висловлень L_1?

5. Сформулюйте метатеорему дедукції та її наслідки.
6. Які похідні правила виведення мають місце в численні висловлень L_1?

ВПРАВИ

2.1 Виписати всі підформули заданої формули: $\left(a \vee b \to c \to \overline{a}\right) \wedge b \wedge c$.

2.2 З'ясувати, чи буде слово формулою, використовуючи аналіз його структури (дерево аналізу): $\left(\left(\left(a \vee b\right) \wedge (c \to \overline{a}\right) \to \overline{b}\right)$.

2.3 Вказати приклади теорем числення висловлень L_1 з виводом довжини n, які одержані з аксіом заданої групи :

a) застосуванням правила підстановки, п=3, група аксіом II;

b) застосуванням правил MP і підстановки п=4, групи аксіом II і IV;

c) застосуванням правил MP і підстановки п=4, групи аксіом III і I.

2.4 Побудуйте формальне доведення теорем теорії L_1, користуючись лише основними правилами виводу:

a) $a \to \left(a \to a\right)$;

b) $a \to a \wedge a$;

c) $a \vee a \to a$;

d) $\overline{b \to a} \to \overline{a}$;

e) $(a \wedge c) \to \overline{\overline{c}}$

f) $a \vee b \to b \vee a$;

g) $a \to \left(b \to \overline{\overline{b}}\right)$;

h) $a \wedge \left(a \to b\right) \to b$.

2.5 Проаналізувати, чи є наступна послідовність формул доведенням для формули $\overline{a \vee b} \to \overline{a} \wedge \overline{b}$?

1) $a \to a \vee b$;

2) $\left(a \to b\right) \to \left(\overline{b} \to \overline{a}\right)$;

3) $\left(a \to \left(a \vee b\right)\right) \to \left(\overline{a \vee b} \to \overline{a}\right)$;

4) $\overline{a \vee b} \to \overline{a}$;

5) $b \to a \vee b$;

6) $\left(b \to a \vee b\right) \to \left(\overline{a \vee b} \to \overline{b}\right)$;

7) $\overline{a \vee b} \to \overline{b}$;

8) $(a \to b) \to ((a \to c) \to (a \to b \wedge c))$;

9) $(\overline{a \vee b} \to \overline{a}) \to ((\overline{a \vee b} \to \overline{b}) \to (\overline{a \vee b} \to \overline{a} \wedge \overline{b}))$;

10) $(\overline{a \vee b} \to \overline{b}) \to (\overline{a \vee b} \to \overline{a} \wedge \overline{b})$;

11) $\overline{a \vee b} \to \overline{a} \wedge \overline{b}$.

2.6 Обґрунтувати вивідність формул із припущень:

a) $a \to (b \to c), \quad b, \quad a \vdash d \to c$;

b) $a \vee b,\ b \to a \vdash a$;

c) $a \to (b \wedge c), \quad a \to b, \quad a \vdash c$;

d) $a \vee b \to c,\ a \to c, b \vdash c$.

2.7 Встановити існування похідних правил виведення з метатеореми 2.2.3.

2.8 Використовуючи метатеорему дедукції, довести:

a) $\vdash (a \to (b \to c)) \to (b \to (a \to c))$;

b) $\vdash ((a \to c) \to a) \to ((a \to c) \to c)$;

c) $\vdash ((a \to c) \to (a \to b)) \to (a \to (c \to b))$;

d) $\vdash (b \to c) \to (b \to (a \vee c))$;

e) $\vdash (\overline{a} \to b) \to (\overline{b} \to a)$.

2.9 Навести змістовне доведення і побудувати формальне доведення теорем:

a) Сума кутів трикутника дорівнює 180^0.

b) Сума кутів опуклого n-кутника дорівнює $180^0(n\text{-}2)$

c) Дві прямі, паралельні третій прямій, паралельні між собою.

d) Діагоналі паралелограма в точці перетину діляться навпіл.

e) Якщо площина перпендикулярна до однієї з двох паралельних прямих, то вона перпендикулярна і до другої прямої.

f) Діагоналі ромба перетинаються під прямим кутом.

3. ЛОГІКА ПРЕДИКАТІВ

3.1. ПРЕДИКАТИ ТА ОПЕРАЦІЇ НАД НИМИ.

Нехай *M* – непорожня підмножина декартового добутку $M_1 \times M_2 \times ... \times M_n$ деяких множин (*n≥1*).

n-місним предикатом, заданим на множині M, називається речення, що містить n змінних $x_1, x_2, ..., x_n$, яке перетворюється на висловлення при підстановці замість цих змінних відповідних конкретних значень $a_1, a_2, ..., a_n$, для яких $(a_1, a_2, ..., a_n) \in M$.

Тобто, на змістовному рівні, n-місний предикат – це відображення $P: M \to \{1,0\}$ від n змінних, яке приймає значення в множинні висловлень.

В подальшому викладі будемо використовувати наступні позначення: $P(x_1, x_2, ..., x_n)$ – предикат; $x_1, x_2, ..., x_n$ – предметні змінні; елементи множин, що їх пробігають предметні змінні – предметні константи; $|P(a_1, a_2, ..., a_n)|$ – значення істинності предиката для відповідного набору елементів.

Відмітимо, що для кожного конкретного набору елементів з множини, на якій задано наш предикат, він перетворюється на висловлення, яке є або істинним (тоді значення предиката для цього набору дорівнює одиниці), або хибним (тоді значення предиката для цього набору буде хибним). Також вкажемо, що не зменшуючи загальності міркувань, будь-яке висловлення будемо вважати *0*-місним предикатом.

Приклад 1. Речення «*x* ділиться на 2 без остачі» є одномісним предикатом на множині натуральних чисел. Позначимо його $P(x)$. Тоді при підстановці замість *x* конкретних натуральних чисел будемо отримувати конкретні висловлення, які будуть або істині, або хибні. Так «4 ділиться на 2 без остачі» – істинне висловлення, і тому $|P(4)| = 1$, а «3 ділиться на 2 без остачі» – хибне висловлення і $|P(3)| = 0$.

Приклад 2. Речення «$x^2 + y^2 = z^2$» є тримісним предикатом на множині цілих чисел. Позначимо його $P(x, y, z)$. Тоді $|P(3,4,5)| = 1$, а $|P(1,1,1)| = 0$.

Предикат $P(x_1, x_2, ..., x_n)$, заданий на множині $M \subset M_1 \times M_2 \times ... \times M_n$, називається:

1) *тотожно істинним*, якщо для будь-якого набору предметних констант $(a_1, a_2, ..., a_n) \in M$ він перетворюється в істинне висловлення, тобто $|P(a_1, a_2, ..., a_n)| = 1$;

2) *тотожно хибним*, якщо для будь-якого набору предметних констант $(a_1, a_2, ..., a_n) \in M$ він перетворюється в хибне висловлення, тобто $|P(a_1, a_2, ..., a_n)| = 0$;

3) *виконуваним*, якщо існує такий набір предметних констант $(a_1, a_2, ..., a_n) \in M$, для якого значення істинності $|P(a_1, a_2, ..., a_n)| = 1$;

4) *спростовним*, якщо існує такий набір предметних констант $(a_1, a_2, ..., a_n) \in M$, для якого значення істинності $|P(a_1, a_2, ..., a_n)| = 0$.

Приклад 3. На множині дійсних чисел предикат «$x \le x+1$» є тотожно істинним, а отже і виконуваним предикатом. На цій же множині предикат «$x \le 0$» є виконуваним і спростовним предикатом одночасно. На множині натуральних чисел предикат «$x+1 \le x$» є тотожно хибним, а отже і спростовним.

Множиною (областю) істинності предиката $P(x_1, x_2, ..., x_n)$, заданого на множині $M \subset M_1 \times M_2 \times ... \times M_n$, називається множина всіх наборів $(a_1, a_2, ..., a_n) \in M$, для кожного з яких $|P(a_1, a_2, ..., a_n)| = 1$. Цю множину надалі позначатимемо M_P. Відмітимо, що $M_P \subset M$.

Приклад 4. Розглянемо предикат «$x \ge 0$» на множині дійсних чисел. Множиною істинності цього предиката буде множина всіх невід'ємних дійсних чисел. Множиною істинності предиката «$x^2 < 0$», заданого на множині дійсних чисел, буде порожня множина, оскільки квадрат будь-якого дійсного числа завжди більший за нуль або дорівнює йому. Множиною істинності предиката «$x < 4$», заданого на множині натуральних чисел буде множина $\{1,2,3\}$.

Тепер розглянемо основні операції над предикатами. Далі будемо вважати, що предикати $P(x_1, x_2, ..., x_n)$ і $Q(x_1, x_2, ..., x_n)$ задано на одній і тій самій множині $M \subset M_1 \times M_2 \times ... \times M_n$.

Запереченням предиката $P(x_1, x_2, ..., x_n)$ називається предикат, заданий на тій же множині, який перетворюється в хибне висловлення для будь-якого набору з множини істинності заданого предиката $P(x_1, x_2, ..., x_n)$, і в істинне – для всіх інших наборів значень предметних змінних, і позначається $\overline{P(x_1, x_2, ..., x_n)}$.

Кон'юнкцією предикатів $P(x_1, x_2, ..., x_n)$ та $Q(x_1, x_2, ..., x_n)$ називається n-місний предикат (заданий на множині $M \subset M_1 \times M_2 \times ... \times M_n$), який перетворюється в істинне висловлення для всіх тих і тільки тих значень змінних, при яких перетворюються в істинні висловлення обидва задані предикати.

Пропонуємо читачеві спробувати самостійно дати означення операцій диз'юнкції, імплікації та еквіваленції. Відмітимо лише, що вони означаються аналогічно до операції кон'юнкції.

Оскільки, в результаті однієї з операцій над предикатами P і Q утвориться новий предикат, то можна говорити про його множину істинності. Очевидно постає запитання, а чи можна якось пов'язати множину істинності результату з множинами істинності вихідних предикатів? Відповідь на це питання дає наступна теорема.

Теорема 3.1.1. Нехай предикати $P(x_1, x_2, ..., x_n)$ і $Q(x_1, x_2, ..., x_n)$ задано на одній і тій же множині $M \subset M_1 \times M_2 \times ... \times M_n$, і нехай предикат $R(x_1, x_2, ..., x_n)$ утворюється внаслідок однієї з логічних операцій, тоді для множини істинності предиката R справедливі наступні рівності:

1) $M_R = M \setminus M_P$, якщо $R = \overline{P}$;
2) $M_R = M_P \cap M_Q$, якщо $R = P \wedge Q$;
3) $M_R = M_P \cup M_Q$, якщо $R = P \vee Q$;
4) $M_R = M \setminus (M_P \setminus M_Q)$, $R = P \to Q$;
5) $M_R = (M_P \cap M_Q) \cup (M \setminus (M_P \cup M_Q))$, якщо $R = P \leftrightarrow Q$.

Доведення цієї теореми нескладне, і випливає з означення логічних операцій над предикатами та означення і властивостей операцій над множинами.

Приклад 5. Розглянемо два одномісні предикати задані на множині натуральних чисел. Предикат $P(x)$ означає «x ділиться на 2», предикат $Q(x)$ – «x ділиться на 3». Множинами істинності цих предикатів очевидно будуть множини $M_P = \{x \mid (x \in N) \wedge (x \vdots 2)\}$ і $M_Q = \{x \mid (x \in N) \wedge (x \vdots 3)\}$. Нехай $R(x)$ результат однієї з логічних операцій, виконаних над предикатами $P(x)$ і $Q(x)$. Тоді множина істинності предикату $R(x)$ буде мати наступний вигляд, в залежності від операції:

1) $M_R = \{x \mid (x \in N) \wedge (x \overline{\vdots} 2)\}$, якщо $R = \overline{P}$;
2) $M_R = \{x \mid (x \in N) \wedge (x \vdots 6)\}$, якщо $R = P \wedge Q$;
3) $M_R = \{x \mid (x \in N) \wedge ((x \vdots 2) \vee (x \vdots 3))\}$, якщо $R = P \vee Q$;
4) $M_R = \{x \mid (x \in N) \wedge ((x \overline{\vdots} 2) \vee (x \vdots 3)\}$, якщо $R = P \to Q$;
5) $M_R = \{x \mid (x \in N) \wedge ((x \vdots 6) \vee (x \overline{\vdots} 2) \wedge (x \overline{\vdots} 3))\}$, якщо $R = P \leftrightarrow Q$.

В логіці предикатів використовують ще так звані операції зв'язування квантором (операції квантифікації або квантування).

Зв'язування квантором загальності за змінною x_i називається логічна операція, яка кожному *n*-місному предикату $P(x_1, x_2, ..., x_n)$, заданому на множині $M \subset M_1 \times M_2 \times ... \times M_n$, ставить у відповідність (*n-1*)-місний предикат, що позначається $\forall x_i P(x_1, .., x_i, ..., x_n)$, і який перетворюється в істинне висловлення при підстановці замість x_k будь-яких конкретних значень $a_k \in M_k$ $(k \neq i, 1 \leq k \leq n)$ тоді і тільки тоді, коли одномісний предикат $P(a_1, .., x_i, ..., a_n)$ тотожньо істинний на множині M_i.

По іншому це можна записати так

$$\left|\forall x_i P(x_1, ..., x_i, ..., x_n)\right| = \begin{cases} 1, \text{ якщо } P(a_1, ..., x_i, ..., a_n) - \text{тотожно істиний предикат}; \\ 0, \text{ якщо } P(a_1, ..., x_i, ..., a_n) - \text{спростовний предикат.} \end{cases}$$

Зв'язування квантором існування за змінною x_i називається логічна операція, яка кожному *n*-місному предикату $P(x_1, x_2, ..., x_n)$, заданому на множині $M \subset M_1 \times M_2 \times ... \times M_n$, ставить у відповідність (*n-1*)-місний предикат, що позначається $\exists x_i P(x_1, .., x_i, ..., x_n)$, і який перетворюється в хибне висловлення при підстановці замість x_k будь-яких конкретних значень $a_k \in M_k$ $(k \neq i, 1 \leq k \leq n)$ тоді і тільки тоді, коли одномісний предикат $P(a_1, .., x_i, ..., a_n)$ тотожно хибний на множині M_i.

Інакше кажучи,

$$\left|\exists x_i P(x_1, ..., x_i, ..., x_n)\right| = \begin{cases} 0, \text{ якщо } P(a_1, ..., x_i, ..., a_n) - \text{тотожно хибний предикат}; \\ 1, \text{ якщо } P(a_1, ..., x_i, ..., a_n) - \text{виконуваний предикат}. \end{cases}$$

Приклад 6. Розглянемо двомісний предикат «$x + y > 1$», заданий на множині натуральних чисел, та зв'яжемо квантором загальності змінну *x*. Отримаємо одномісний предикат «$\forall x\, (x + y > 1)$», який буде тотожно істинним для будь-якого значення *y*.

Відмітимо, що якщо є *n*-місний предикат, то його змінні можна по черзі зв'язувати одним із кванторів. Якщо зв'язати кванторами всі змінні деякого предиката, то отримаємо висловлення.

Приклад 7. Як приклад розглянемо предикат «$x + y < z$», заданий на множині дійсних чисел. Зв'яжемо всі його змінні кванторами так «$\forall x \forall y \exists z\, (x + y < z)$» і одержимо істинне висловлення. Дійсно, для будь-яких двох дійсних чисел x і y можна вказати таке дійсне число z, що $x+y<z$ (для цього достатньо покласти $z=x+y+1$).

Перед введенням поняття формули логіки предикатів розглянемо ще деякі важливі поняття, а саме поняття області дії квантора, вільного і зв'язаного входжень предметної змінної.

Область дії квантора – це предикат, до якого відноситься цей квантор.

Входження предметної змінної x в даний предикат називається *зв'язаним*, якщо x є змінною квантора або знаходиться в області дії квантора за цією змінною. В протилежному випадку входження предметної змінної називається *вільним*.

Предметна змінна називається *вільною* в даному предикаті, якщо всі її входження в цей предикат вільні; *зв'язаною* – якщо всі її входження зв'язані; *частково вільною або частково зв'язаною* – якщо є вільні і зв'язані входження цієї змінної в предикат.

Приклад 8. Розглянемо предикат «$\exists x \forall y (x \le y^2) \Rightarrow (x=0)$». Областю дії квантора існування буде предикат «$\forall y (x \le y^2)$», областю дії квантора загальності буде предикат «$x \le y^2$», змінна y є зв'язаною змінною, а змінна x – частково зв'язаною, оскільки має одне вільне та одне зв'язане входження.

Якщо одномісний предикат $P(x)$ задано на скінченній множині елементів $M=\{c_1, c_2, ..., c_k\}$, то операція зв'язування квантором загальності $\forall x P(x)$ рівносильна кон'юнкції $P(c_1) \wedge P(c_2) \wedge ... \wedge P(c_k)$, а операція зв'язування квантором існування $\exists x P(x)$ рівносильна диз'юнкції $P(c_1) \vee P(c_2) \vee ... \vee P(c_k)$.

При формулюванні тверджень мовою логіки предикатів часто зустрічаються речення чотирьох типів, які в арістотелевій логіці називаються *категоричними судженнями* і мають зміст та символічний запис:

A: загальностверджувальне судження *"всі S суть P"* (всі елементи *x*, які мають властивість *S*, мають і властивість *P*) - $\forall x (S(x) \to P(x))$;

E: загальнозаперечувальне судження "*будь-яке S не є P*" (будь-який елемент *x*, який має властивість *S*, не має властивості *P*)- $\forall x (S(x) \to \overline{P(x)})$;

I: частково стверджувальне судження *"деякі S суть P"* (деякі елементи *x*, які мають властивість *S*, мають і властивість *P*) - $\exists x (S(x) \wedge P(x))$;

O: частково заперечувальне судження *"деякі S не є P"* (деякі елементи *x*, які мають властивість *S*, не мають властивості *P*) - $\exists x (S(x) \wedge \overline{P(x)})$;

Комбінуючи речення *A-O*, можна записувати у символічній формі досить складні твердження.

Запитання для самоконтролю

1. Що називається n-місним предикатом, заданим на множині M=$M_1 \times M_2 \times ... \times M_n$? Як позначаються предикати?
2. Що таке предметні змінні, предметні константи, значення істинності предиката для відповідного набору?
3. Який предикат називається: а)тотожно істинним, б)тотожно хибним, в)виконуваним, г)спростовним? Наведіть приклади предикатів кожного типу.
4. Що називається областю істинності предиката $P(x_1, x_2, ..., x_n)$ заданого на множині $M \subset M_1 \times M_2 \times ... \times M_n$?
5. Як визначаються операції заперечення, кон'юнкція, диз'юнкція, імплікація та еквіваленція в логіці предикатів?
6. Що називається зв'язуванням квантором загальності за змінною x_i предиката $P(x_1, x_2, ..., x_n)$, заданого на множині $M \subset M_1 \times M_2 \times ... \times M_n$?
7. Що називається зв'язуванням квантором існуванняза змінною x_i предиката $P(x_1, x_2, ..., x_n)$, заданого на множині $M \subset M_1 \times M_2 \times ... \times M_n$?
8. Що таке область дії квантора? Які змінні називаються зв'язаними, вільними, частково зв'язаними або частково вільними?
9. Як пов'язані операції зв'язування квантором загальності та кон'юнкція в логіці предикатів?
10. Як пов'язані операції зв'язування квантором існування та диз'юнкція в логіці предикатів?

ВПРАВИ

3.1. Визначити, які з наступних речень будуть предикатами та вказати їх місність:

a) 4 і 6 діляться на 3 без остачі;

b) *x* – парне число;

c) *x* і *y* чотирикутники;

d) *f(x)* – диференційовна функція;

e) кожен день неділі – середа;

f) сьогодні четвер і буде хороша погода;

g) у будь-якому трикутнику можна провести три висоти.

3.2. Вказати приклади тотожно істинних, тотожно хибних, виконуваних та спростовних предикатів, заданих на множині:

a). цілих чисел;

b). простих чисел;

c). {-1,0,1}.

3.3. Для кожного з наступних висловлень знайти предикати, які перетворюються в дане висловлення при заміні предметних змінних їх деякими значеннями з області задання предиката:

a). 6>2+3;

b). гори Карпати знаходяться в Україні;

c). через точку A(1;0) можна провести пряму, паралельну до прямої b.

3.4. Визначити які з входжень змінних у формулу є вільними, а які зв'язаними:

a) $\forall y(\exists x P(x,y) \vee Q(y,z))$;

b) $\exists x(P(x,y) \to \forall y R(y)) \wedge \forall y \exists z Q(x,y,z)$;

c) $(\exists y P(x,y) \wedge \forall x R(x,y)) \to \overline{\exists y Q(x,y)}$;

d) $(\forall x P(x,y) \wedge \exists z Q(x,y,z)) \wedge R(x)$.

3.5. Визначити множини істинності наступних предикатів:

a) *P(x)*= «$|x| > 2$», *M* – множина дійсних чисел;

b) *P(x)*= «x парне і просте число», *M* – множина натуральних чисел;

c) *P(x)*= «x взаємно просте з 5», *M* – множина цілих чисел;

d) *P(x)*= «x>x+1», *M* – множина від'ємних чисел.

3.6. Визначити множини істинності наступних предикатів, заданих на множині M, та зобразити їх на площині:

a) $P(x,y) \wedge Q(x,y)$, де *P(x,y)* = «$|x+y| > 2$», *Q(x,y)* = «$x^2 + y^2 \le 9$», M –множина дійсних чисел;

b) $\overline{Q(x,y)}$, де *Q(x,y)* = «$x^2 - y^2 > 10$», M – множина цілих чисел;

c) $P(x,y) \wedge Q(x,y)$, де P(x,y) = «$x = y$», *Q* (x,y) = «$x = e^{\ln(y)}$», M – множина дійсних чисел;

d) $P(x,y) \vee Q(x,y)$, де P(x,y) = «$x > y$», *Q* (x,y) = «$x + y > 3$», M – множина дійсних чисел;

e) $P(x,y) \to Q(x,y)$, де P(x,y)= «$x^3 + y = 1$», *Q* (x,y)= «$x - 3 > y$», M – множина дійсних чисел;

f) $Q(x,y) \to P(x,y)$, де P(x,y)= «$x^3 + y = 1$»,Q (x,y)= «$x - 3 > y$», M – множина дійсних чисел;

g) $P(x,y) \leftrightarrow Q(x,y)$, де P(x,y)= «$x^3 + y = 1$»,$Q$ (x,y)= «$x^2 - y^2 > 10$», M – множина дійсних чисел.

3.7. Нехай на множині людей задано предикати *P(x,y)* = « *x* батько *y*», *Q(x,y)*= «*x* мати *y*», *R(x,y)* = «*x* син *y*» та *S(x,y)* = «*x* дочка *y*». Виразити через них наступні предикати:

a) «*x* брат *y*»;

b) «*x* тітка *y*»;

c) «*x* бабуся *y* з боку матері»;

d) «*x* і *y* - двоюрідні сестри»;

e) «*x* і *y* - внуки *z*»;

3.8.Записати речення у символічній формі, увівши потрібні предикати*:*

a) жодне парне число, більше 2, не просте;

b) добуток двох довільних чисел не є простим числом;

c) добуток двох парних чисел є парне число;

d) існують неперервні функції, які не диференційовані;

e) кожне дійсне число, за виключенням нуля, має обернене;

f) деякі автобуси не зупиняються на цій зупинці’

g) кожний, в кому є упертість, може вивчити логіку;

h) всі судді юристи, але не всі юристи – судді;

i) ви можете обманювати декого весь час, ви можете обманювати всіх деякий час, але ви не можете обманювати всіх і весь час.

3.9. Предикати *P(x,y)*, *R(x,y)* та *Q(x)* за дано на множині $\{a,b,c\}$ таблицями значень:

P(x,y)

x\y	*a*	*b*	*c*
a	1	1	0
b	0	1	0
c	0	1	1

R(x,y)

x\y	*a*	*b*	*c*
a	1	0	1
b	1	1	1
c	0	0	1

x	*Q(x)*
a	1
b	0
c	1

Побудувати таблиці значень предикатів:

a) $Q(x) \to (\exists y \forall x P(x,y))$;

b) $\exists x P(x,y) \vee \exists y R(x,y)$;

c) $Q(x) \leftrightarrow \exists x(P(x,y) \wedge R(x,y))$;

d) $\forall x \exists y(Q(x) \wedge P(x,y) \wedge R(x,y))$;

e) $\exists y(P(x,y) \leftrightarrow R(x,y)) \wedge Q(x)$.

3.10. Для предикатів P(x,y) та Q(x), заданих на множині $\{a,b,c\}$, скласти таблиці значень так, щоб предикат R(x,y) був:

a). тотожно істинним, якщо $R(x,y) = Q(x) \to \forall y P(x,y)$;

b). тотожно хибним, якщо $R(x,y) = \forall x Q(x) \vee P(x,y)$;

c). виконуваним, якщо $R(x,y) = \forall x Q(x) \wedge \exists y P(x,y) \to \overline{Q(x)}$.

3.11. Студент вирішив кожну прослухану лекцію повторювати дома. В кінці семестру виявилось що він не виконав свого рішення. Запишіть цей факт мовою логіки предикатів.

3.12. Випадок в магазині. Юнак вирішив купити туфлі в подарунок брату але забув розмір його ноги. Тоді продавець сказав йому: «В нашому магазині для будь-якої ноги знайдуться туфлі підходящого розміру». На що юнак відповів: «Тоді дайте мені туфлі, що підходять до будь-якої ноги». Чи правильно юнак зрозумів продавця?

3.13. На множині натуральних чисел з нулем вибрано такі атомарні предикати:

$$S(x_1,x_2,x_3) = \begin{cases} 1, & x_1 + x_2 = x_3, \\ 0, & x_1 + x_2 \neq x_3; \end{cases}$$

$$D(x_1,x_2,x_3) = \begin{cases} 1, & x_1 x_2 = x_3, \\ 0, & x_1 x_2 \neq x_3. \end{cases}$$

Виразити через них такі предикати:

a). $x_1 \geq x_2$;

b). x_1 – просте число;

c). $x_1 = 0$;

d). x_1 – НСД чисел x_2 та x_3.

3.2. ФОРМУЛИ ЛОГІКИ ПРЕДИКАТІВ. ІНТЕРПРЕТАЦІЯ ФОРМУЛ. ВИКОНУВАНІ ТА ЛОГІЧНО ЗАГАЛЬНОЗНАЧУЩІ ФОРМУЛИ

Алфавітом логіки предикатів є наступні категорії символів:

1) предметні змінні;

2) предметні константи;

3) функціональні символи f_i^n (i – порядковий номер, n – арність, кількість аргументів);

4) предикатні символи P_i^n (i – порядковий номер, n – місність);

5) символи логічних операцій: $\overline{}, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$;

6) допоміжні символи: (,).

Слова, які записані за допомогою алфавіту, поділяються на терми та формули. *Термами* є будь-яка предметна змінна або константа, а також значення функціонального символу для набору, кожен елемент якого є термом. Предикатні символи, в застосуванні їх до термів, дають *елементарні формули* логіки предикатів.

Формулами логіки предикатів є:

1) будь-які елементарні формули;

2) слова виду: $\overline{(\alpha)}, (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \rightarrow \beta), (\alpha \leftrightarrow \beta)$, де α і β – формули логіки предикатів;

3) якщо α формула логіки предикатів, а x_i вільна змінна цієї формули, то слова $(\forall x_i(\alpha))$ і $(\exists x_i(\alpha))$ також є формулами логіки предикатів.

4) всі інші слова, крім тих, які утворюються за правилами 1)-3), не є формулами логіки предикатів.

Приклад 1. Перевірити, чи утворюють наступні слова формули логіки предикатів: «$\exists x_1\left(\left(P_1^1(x_1)\right) \rightarrow \left(\forall x_1\left(P_2^2(x_1,x_2)\right)\right)\right)$», «$\left(\overline{\overline{P_1^2(x_1,x_2)} \rightarrow \left(P_2^2(x_1,x_2)\right)}\right)$».

Слово $\exists x_1\left(\left(P_1^1(x_1)\right) \rightarrow \left(\forall x_1\left(P_2^2(x_1,x_2)\right)\right)\right)$ не є формулою логіки предикатів, оскільки $\left(P_1^1(x_1)\right) \rightarrow \left(\forall x_1\left(P_2^2(x_1,x_2)\right)\right)$ є формулою згідно пунктів 1)-3), але в цій формулі змінна x_1 є частково вільною, а тому навішування квантора існування не відповідає пункту 3) і, згідно 4), слово $\exists x_1\left(\left(P_1^1(x_1)\right) \rightarrow \left(\forall x_1\left(P_2^2(x_1,x_2)\right)\right)\right)$ не утворює формулу логіки предикатів.

Слово $\left(\overline{\overline{P_1^2(x_1,x_2)} \rightarrow \left(P_2^2(x_1,x_2)\right)}\right)$ утворює формулу логіки предикатів. Дійсно, слова $P_1^2(x_1,x_2)$ і $P_2^2(x_1,x_2)$ є формулами згідно пункту 1). А слова $\overline{P_1^2(x_1,x_2)}$, $\overline{P_1^2(x_1,x_2)} \rightarrow \left(P_2^2(x_1,x_2)\right)$ і $\left(\overline{\overline{P_1^2(x_1,x_2)} \rightarrow \left(P_2^2(x_1,x_2)\right)}\right)$ є формулами логіки предикатів згідно пункту 2). Отже формула $\left(\overline{\overline{P_1^2(x_1,x_2)} \rightarrow \left(P_2^2(x_1,x_2)\right)}\right)$ є формулою логіки предикатів.

Формула, яка не має вільних входжень предметних змінних, називається замкнутою. В протилежному випадку формула є відкритою.

Інтерпретацією формули логіки предикатів $\alpha(x_1,...,x_n)$ називається система, яка складається з непорожньої множини $M \subset M_1 \times M_2 \times ... \times M_n$, яку називають областю інтерпретації, і деякої відповідності, яка кожному предикатному символу P_i^n ставить у відповідність певний *n*-місний предикат, заданий на множині *M*, кожному функціональному символу f_i^n – деяку *n*-арну операцію, кожній константі a_i – деякий конкретний елемент із M_i.

Приклад 2. Побудувати деяку інтерпретацію для формули $\exists x P_1^3\left(f_1^1(x), y, z\right)$.

Наша формула містить тримісний предикат, один функціональний символ, дві вільні та одну зв'язану змінну. За область інтерпретації оберемо множину всіх дійсних чисел. Функціональному символу поставимо у відповідність унарну операцію піднесення до квадрата $f_1^1(x) = x^2$. Предикатний символ позначимо тримісним предикатом $P_1^3(x,y,z) =$ «$x = y + z$». В цій інтерпретації ми отримали формулу $\exists x\left(x^2 = y + z\right)$, яка буде виконуватись на тих наборах дійсних чисел, для яких $y + z \geq 0$ і $x = \pm\sqrt{z+y}$.

Формули логіки предикатів в деякій фіксованій інтерпретації поділяються на: *істинні* (виконуються для будь-якого набору з області інтерпретації), *хибні* (не виконуються на жодному наборі з області), *виконувані* (виконуються хоч на одному наборі) та *спростовні* (не виконуються хоч на одному наборі).

Якщо формула логіки предикатів істинна в будь-якій інтерпретації, то така формула називається *загальнозначущою або логічно загальнозначущою (ЛЗЗ)*. Логічно загальнозначущу формулу називають *тавтологією логіки предикатів*. Якщо формула логіки предикатів хибна в будь-якій інтерпретації, то вона називається *тотожно хибною або суперечністю логіки предикатів*. Якщо формула логіки предикатів виконувана (спростовна) хоча б в одній інтерпретації, то така формула називається *виконуваною (спростовною)*.

Інтерпретація називається *моделлю* для даної множини *Г* формул логіки предикатів, якщо кожна формула з *Г* істинна в даній інтерпретації.

Якщо ми маємо деяку формулу алгебри висловлення $A(a_1, a_2, ..., a_n)$, то при застосуванні підстановки $S_{a_1,a_2,...,a_n}^{\beta_1,\beta_2,...,\beta_n}(A)$, де $\beta_1, \beta_2, ..., \beta_n$ – довільні формули логіки предикатів, ми отримаємо деяку формулу логіки предикатів, яку називають *окремим випадом формули алгебри висловлень*. Це правило дає можливість на основі тотожно істинних формул алгебри висловлень будувати, використовуючи правило підстановки, загальнозначущі формули логіки предикатів – *окремі випадки тавтологій алгебри висловлень*.

Приклад 3. Формула логіки предикатів

$$\left(\exists x P(x) \to \forall y (Q(x,y)\right) \leftrightarrow \left(\overline{\forall y Q(x,y)} \to \overline{\exists x P(x)}\right)$$

є загальнозначущою, оскільки одержана підстановкою

$$S_{a\quad b}^{\exists x P(x)\quad \forall y Q(x,y)}\left((a \to b) \leftrightarrow \left(\overline{b} \to \overline{a}\right)\right),$$

застосованою до формули алгебри висловлень $(a \to b) \leftrightarrow (\overline{b} \to \overline{a})$, що є законом контрапозиції.

Приклад 4. Чи буде формула логіки предикатів $\alpha(x) = \forall x \left(P(x) \to Q(x)\right) \wedge \overline{Q(x)}$ виконуваною? ЛЗЗ?

Знайдемо умови, які повинні задовольняти предикати в інтерпретації. Припустимо, що існує інтерпретація φ на множині M: $\varphi(\alpha(x)) = 1$. Тобто $\exists a_0 \in M$ і існують інтерпретації предикатних символів $P(x)$ і $Q(x)$ - $P_0(x)$ і $Q_0(x)$ - такі, що:

$$\left|\forall x \left(P_0(x) \to Q_0(x)\right) \wedge \overline{Q_0(a_0)}\right| = 1.$$

Це рівносильно системі умов:

$$\begin{cases} \left|\forall x \left(P_0(x) \to Q_0(x)\right)\right| = 1, \\ \left|\overline{Q_0(a_0)}\right| = 1. \end{cases}$$

Останню систему можна переписати так:

$$\begin{cases} \forall x \in M \ \left|P_0(x) \to Q_0(x)\right| = 1, \\ \left|\overline{Q_0(a_0)}\right| = 1. \end{cases}$$

Умови системи не суперечать одна одній. Тобто: $Q(x)$ повинен бути спростовним і на всіх значеннях множини інтерпретації, де $Q(x)$ набуває хибних значень, $P(x)$ також повинен набувати хибних значень, або бути тотожно хибним.

Наприклад, нехай $Q(x)$=«$x \vdots 3$» і $P(x)$=«$x \vdots 6$» на множині $M = Z$. Одержимо формулу $\forall x \left((x \vdots 6) \to (x \vdots 3)\right) \wedge (x \overline{\vdots} 3)$, яка набуває істинних значень на всіх цілих числах $x \notin \{3k \mid k \in Z\}$. Таким чином $\alpha(x)$ - виконувана.

Можна розглянути інтерпретацію нашої формули на одноелементній множині $M = \{a\}$. На цій множині $\forall x \left(P(x) \to Q(x)\right)$ та $\overline{Q(x)}$ перейдуть відповідно в $P(a) \to Q(a)$ і $\overline{Q(a)}$, які позначимо $p \to q$ і $\overline{q}$. Тоді матимемо формулу алгебри висловлень, яка є спростовною. Дійсно:

$$(p \to q) \wedge \overline{q} \equiv (\overline{p} \vee q) \wedge \overline{q} \equiv \overline{p} \wedge \overline{q}.$$

Остання формула алгебри висловлень є спростовною. Отже, задана формула логіки предикатів $\alpha(x)$ не є істинною в даній інтерпретації, і, як наслідок, не є ЛЗЗ.

Приклад 5. Довести, що формула логіки предикатів $\exists xP(x) \to (\forall xQ(x) \to \exists xP(x))$ є ЛЗЗ.

Припустимо, що наша формула є спростовною. Тобто існує така інтерпретація на деякій множині *M*, на якій предикатні символи $P(x)$ і $Q(x)$ мають інтерпретації $P_0(x)$ і $Q_0(x)$, і $|\exists xP_0(x) \to (\forall xQ_0(x) \to \exists xP_0(x))| = 0$.

Остання рівність рівносильна системі

$$\begin{cases} |\exists xP_0(x)| = 1, \\ |\forall xQ_0(x) \to \exists xP_0(x)| = 0. \end{cases}$$

Використовуючи означення операцій квантування та імплікації прийдемо до рівносильної системи

$$\begin{cases} \exists x_0 \in M\ |P_0(x_0)| = 1, \\ \forall x \in M\ |Q_0(x)| = 1, \\ \forall x \in M\ |P_0(x)| = 0, \end{cases}$$

яка містить суперечність. Отже наше припущення про спростовність формули $\exists xP(x) \to (\forall xQ(x) \to \exists xP(x))$ є невірним. А тому вона є логічно загальнозначущою.

<u>Запитання для самоконтролю</u>

1. Які типи символів становлять алфавіт логіки предикатів?
2. Що називається термом?
3. Що називається формулою логіки предикатів? Що таке атомарна (елементарна) формула?
4. Яка формула логіки предикатів називається замкненою, відкритою?
5. Що називається інтерпретацією формули логіки предикатів $\alpha(x_1,...,x_n)$?
6. Які типи формул логіки предикатів в фіксованій інтерпретації існують?
7. Яка формула називається логічно загальнозначущою? Наведіть приклад ЛЗЗ формули.
8. Яка формула називається виконуваною? Наведіть приклад.
9. Що таке модель множини формул?
10. Що називають окремим випадом формули алгебри висловлень?

<u>ВПРАВИ</u>

3.14. Перевірити чи утворюють наступні слова формули логіки предикатів:

a) $\forall x\big(\exists x\big(P(x,y)\to\forall yR(y)\big)\wedge\forall y\exists zQ(x,y,z)\big)$;

b) $\big(\forall xP(x,y)\wedge\exists zQ(x,y,z)\big)\wedge R(x)$;

c) $\exists x\big(R(y)\to\forall x\forall z\forall yP(x,y,z)\big)$;

d) $\exists x\big(P(x)\vee Q(x)\big)\leftrightarrow\big(\exists xP(x)\vee\exists xQ(x)\big)$.

3.15. Записати наведені нижче математичні твердження за допомогою формул логіки предикатів:

a) існують такі дійсні числа x, y, z, що $x^2=y^2+z^2$;

b) будь-який опуклий чотирикутник має площу;

c) якщо кожна з двох прямих паралельна третій, то вони паралельні між собою;

d) для довільного дійсного числа x: $-1\le\sin(x)\le 1$;

e) існують три точки, що не належать одній прямій.

3.16. Побудувати інтерпретації для наступних формул:

a) $\forall x\forall y\big(P_1^2(x,y)\to P_1^2(y,x)\big)$;

b) $\forall x\forall y\big(P_1^2(x,y)\vee P_1^2(y,x)\big)$;

c) $\forall x\exists yP_1^2(f_1^1(x),f_2^1(y))$;

d) $\forall xP_1^2(x)\to\exists y\exists zP_2^2(x,f_1^1(y),f_2^1(z))$;

e) показати, що при інтерпретації формули $\exists xP(x)\to\forall xP(x)$ на довільній одноелементній множині завжди отримується істинне висловлення, а на двоелементній – не завжди.

3.17. Записати деяку інтерпретацію формули над множиною $\{a,b,c\}$ і записати таблицю значень отриманого предиката:

a). $\big(\exists yP(x,y)\wedge\forall xQ(x,y)\big)\to\exists yR(x,y)$;

b). $\overline{\big(\forall xP(x,z)\vee\exists yQ(y,x)\big)}\leftrightarrow R(x,z)$;

c). $\forall xP(x,y)\wedge\exists zQ(x,y,z)\vee R(x)$.

3.18. Навести приклад формули F логіки предикатів, для якої виконується наступна умова:

a). F є тотожно істиною на всіх скінченних множинах і виконуваною на всіх нескінченних;

b). F є виконуваною на деякій нескінченній множині і тотожно хибною на всіх скінченних множинах;

c). F виконувана на скінченних множинах з парною кількістю елементів і тотожно хибна на всіх інших скінченних множинах.

3.19. Перевірити, чи буде формула логіки предикатів логічно загальнозначущою:

a) $P(x) \to \exists y Q(y)$;

b) $\forall x(P(x) \to Q(y)) \leftrightarrow (\exists x P(x) \to Q(y))$;

c) $\exists x(P(x,y) \to Q(y)) \to \forall z Q(z)$

d) $(\overline{\forall x P(x)}) \leftrightarrow \exists x(\overline{P(x)})$;

e) $\exists x(P(x) \vee Q(x)) \leftrightarrow (\exists x P(x) \vee \exists x Q(x))$;

f) $(\exists y P(x,y) \wedge \forall x Q(x,y)) \to \exists y R(x,y)$

g) $\forall x(P(x) \to \overline{Q(x)}) \to (\overline{\forall x P(x) \wedge \exists x Q(x)})$;

h) $\exists x(\overline{P(x,y)} \to Q(x)) \leftrightarrow \forall y P(x,y)$

i) $(P(x) \vee \forall y Q(y)) \to \forall y(P(x) \vee Q(y))$.

3.20. Перевірити, чи буде виконуваною формула логіки предикатів:

a) $\exists x \forall y(P(x,y) \to \forall z Q(x,y,z))$;

b) $\exists x P(x) \to P(y)$;

c) $\exists x \exists y(P(x) \wedge \overline{P(y)})$;

d) $\exists x(P(x,y) \leftrightarrow Q(y))$;

e) $\exists x(\overline{P(x,y)} \to Q(x)) \leftrightarrow \forall y P(x,y)$;

f) $\exists x \forall y(P(x,y) \to \forall z Q(x,y,z))$;

g) $\forall x P(x,y) \wedge \exists z Q(x,y,z) \vee R(x)$

h) $\exists x(P(x,y) \to Q(y)) \to \forall z Q(z)$;

i) $\exists x(P(x) \vee Q(x)) \leftrightarrow (\exists x P(x) \vee \exists x Q(x))$

j) $\forall x \exists y P(x,y) \leftrightarrow \exists x \forall y P(x,y)$.

3.3. ВІДНОШЕННЯ РІВНОСИЛЬНОСТІ В ЛОГІЦІ ПРЕДИКАТІВ. ЗВЕДЕНА ТА ПРЕНЕКСНА НОРМАЛЬНІ ФОРМИ

В логіці предикатів, як і в алгебрі висловлень, введемо поняття логічного наслідку та задамо відношення рівносильності на множині формул логіки предикатів.

Формула β називається *логічним наслідком* множини формул $\{\alpha_1, \alpha_2, ..., \alpha_n\}$, якщо вона виконується на всіх наборах елементів з області довільної інтерпретації, на яких виконується кожна формула $\alpha_i, (1 \le i \le n)$. Позначення: $\alpha_1, \alpha_2, ..., \alpha_n \models \beta$.

Якщо множина складається лише з однієї формули $\{\alpha\}$, то кажуть що формула β є логічним наслідком формули α.

Найбільш часто вживані схеми логічного висновку - *силогізми Арістотеля*. Це схеми, що складаються з трьох простих висловлень типуА, Е,І та О, з яких перші два - посилки, а третє - висновок. У кожному з силогізмів розглядаються три властивості (предикати) S, M та P. Перша (велика) посилка пов'язує M і P, друга (мала) пов'язує M і S, а висновок пов'язує S і P. У залежності від положення M у посилках, розрізняють чотири фігури силогізму:

I	II	III	IV
$MxP,\ SyM \models SzP$	$PxM,\ SyM \models SzP$	$MxP,\ MyS \models SzP$	$PxM,\ MyS \models SzP$

Тут буквами x, y та z позначено одне з чотирьох висловлень типу A, E, I чи O, що пов'язує властивості S, M та P. Отже, у кожній фігурі є 4·4·4 = 64 модуси силогізму. Всього різних модусів силогізму 256. Арістотель встановив, що правильними з них є лише 19 (по 4 у I та II фігурах, 6 у III і 5 у IV фігурі). Силогізми позначають трьома літерами (AAA, EIO, AEE і т.п.) за виглядом висловлень, які пов'язують властивості. Для полегшення запам'ятовування правильних силогізмів кожному присвоюють ім'я (Barbara, Ferio, Camestresi т.п.) - латинське слово, яке містить тільки ті голосні букви, що входять до позначення силогізму. Наприклад, Barbara це силогізм AAA першої фігури MAP, SAM⊨SAP, який можна прочитати «Всі M ∈ P. Всі S∈M. Отже, всі S∈P» або записати формулами логіки предикатів $\forall x(M(x) \to P(x)), \forall x(S(x) \to M(x)) \models \forall x(S(x) \to P(x))$. Прикладом такого судження може бути: «Всі ромби - паралелограми. Всі квадрати - ромби. Отже, всі квадрати — паралелограми».

Дві формули логіки предикатів називаються *рівносильними*, якщо перша є логічним наслідком другої і навпаки. Позначення: $\alpha \equiv \beta$.

Теорема 3.3.1. Якщо α і β – формули логіки предикатів, то:

1) формула β є логічним наслідком формули α тоді і тільки тоді, коли $\alpha \to \beta$ – загальнозначуща формула;

2) якщо формула β є логічним наслідком формули α, і формула α в даній інтерпретації істинна, то і формула β є істинною в даній інтерпретації;

3) формули α і β рівносильні тоді і тільки тоді, коли $\alpha \leftrightarrow \beta$ – загальнозначуща формула.

Відмітимо, що для формул логіки предикатів мають місце *всі рівносильності алгебри висловлень*.

Окремим випадком формули А алгебри висловлень називається формула логіки предикатів, одержана з А підстановкою замість пропозиційних змінних довільних формул логіки предикатів.

Окремий випадок будь-якої тавтології алгебри висловлень є загальнозначущою формулою (тавтологією) логіки предикатів. Наприклад, формула $\overline{\overline{\forall x A(x)}} \leftrightarrow \forall x A(x)$ тавтологія, бо є окремим випадком тавтології $\overline{\overline{p}} \leftrightarrow p$. Аналогічно, рівносильність $\overline{\overline{\forall x A(x)}} \equiv \forall x A(x)$ формул логіки предикатів є окремим випадком закона подвійного заперечення алгебри висловлень $\overline{\overline{p}} \equiv p$.

Але крім цього є ще ряд рівносильностей, що стосуються операцій навішування кванторів істинності та загальності.

Теорема 3.3.2. Мають місце наступні рівносильності формул логіки предикатів:

1) $\forall x_1 P(x_1) \equiv \forall x_2 P(x_2), \exists x_1 P(x_1) \equiv \exists x_2 P(x_2)$ (перейменування зв'язаних змінних);

2) $\overline{\forall x P(x)} \equiv \exists x \overline{P(x)}, \overline{\exists x P(x)} \equiv \forall x \overline{P(x)}$ (закони де Моргана для кванторів);

3) $\forall x P_1(x) \wedge \forall x P_2(x) \equiv \forall x \big(P_1(x) \wedge P_2(x)\big),\ \exists x P_1(x) \vee \exists x P_2(x) \equiv \exists x \big(P_1(x) \vee P_2(x)\big)$;

4) $\forall x P_1(x) \wedge P_2 \equiv \forall x \big(P_1(x) \wedge P_2\big),\ \exists x P_1(x) \wedge P_2 \equiv \exists x \big(P_1(x) \wedge P_2\big)$;

5) $\forall x P_1(x) \vee P_2 \equiv \forall x \big(P_1(x) \vee P_2\big),\ \exists x P_1(x) \vee P_2 \equiv \exists x \big(P_1(x) \vee P_2\big)$;

Рівносильності 3)–5) мають назву пронесення кванторів через кон'юнкцію та диз'юнкцію.

6) $\forall x P_1(x) \Rightarrow P_2 \equiv \exists x \big(P_1(x) \Rightarrow P_2\big),\ \exists x P_1(x) \Rightarrow P_2 \equiv \forall x \big(P_1(x) \Rightarrow P_2\big)$;

7) $P_2 \Rightarrow \forall x P_1(x) \equiv \forall x \big(P_2 \Rightarrow P_1(x)\big),\ P_2 \Rightarrow \exists x P_1(x) \equiv \exists x \big(P_2 \Rightarrow P_1(x)\big)$.

Рівносильності 6)–7) мають назву пронесення кванторів через імплікацію. Відмітимо наступне: рівносильності 4)-7) виконуються при умові, що P_2 не містить вільних входжень змінної *x*.

Якщо виконуємо перехід від однієї формули до рівносильної їй формули, то такий перехід називається *рівносильним перетворенням*. Перетворюючи формулу за допомогою рівносильних перетворень, ми будемо або спрощувати її, або отримувати іншу, рівносильну їй формулу спеціального виду. Нижче ми розглянемо формули двох спеціальних видів – зведені та пренексні форми логіки предикатів.

Зведеною формою (ЗФ) для даної формули логіки предикатів називається така рівносильна їй формула, яка або елементарна, або містить лише операції заперечення, кон'юнкції, диз'юнкції, навішування кванторів існування та загальності, причому заперечення стосуються лише елементарних підформул даної формули**.**

В логіці предикатів має місце наступна теорема.

Теорема 3.3.3. Для кожної формули логіки предикатів існує зведена форма.

Приклад 1. Знайти ЗФ для формули $\forall x_1 P_1(x_1, x_2) \to \exists x_2 P_2(x_2)$.

Виконаємо наступні рівносильні перетворення:

$\forall x_1 P_1(x_1, x_2) \to \exists x_2 P_2(x_2) \equiv \overline{\forall x_1 P_1(x_1, x_2)} \vee \exists x_2 P_2(x_2) \equiv$
$\exists x_1 \overline{P_1(x_1, x_2)} \vee \exists x_2 P_2(x_2).$

Пренексною нормальною формою (ПНФ) або випередженою нормальною формою (ВНФ) для даної формули логіки предикатів називається така її зведена форма, яка або не має операцій квантування, або вони виконуються останніми. Тобто ПНФ деякої формули логіки предикатів $\alpha(x)$ має вигляд: $\Theta x_1 \Theta x_2 ... \Theta x_k \beta$, $\Theta \in \{\forall, \exists\}$, де β не містить операцій квантування і називається матрицею ПНФ

Теорема 3.3.4. Для кожної формули логіки предикатів існує пренексна нормальна форма.

Приклад 2. Знайти ПНФ (ВНФ) для формули $\forall x_1 P_1(x_1, x_2) \to \exists x_2 P_2(x_2)$.

В попередньому прикладі ми знайшли ЗФ для даної формули. Застосуємо до неї закон пронесення квантора існування і отримаємо формулу $\exists x_1 \left(\overline{P_1(x_1, x_2)} \vee \exists x_2 P_2(x_2)\right)$, яка ще не утворює ПНФ. У підформулі $\exists x_2 P_2(x_2)$, згідно рівносильності (1) виконаємо перейменування зв'язаної предметної змінної x_2 в нову предметну змінну x_3. Отримаємо нову формулу $\exists x_1 \left(\overline{P_1(x_1, x_2)} \vee \exists x_3 P_2(x_3)\right)$. Тепер, згідно рівносильності (5), отримаємо $\exists x_1 \exists x_3 \left(\overline{P_1(x_1, x_2)} \vee P_2(x_3)\right)$. Неважко переконатися, що це і є ПНФ для формули $\forall x_1 P_1(x_1, x_2) \to \exists x_2 P_2(x_2)$.

При аналізі формул логіки предикатів на виконуваність зручно мати формулу, яка не містить кванторів існування, а її матриця представлена у кон'юнктивній нормальній формі (кнф). Вилучення кванторів існування із префікса ПНФ проводять за допомогою введення *сколемівських сталих* та *сколемівських функцій* за правилом:

1. Знайти перший зліва на право квантор існування. Якщо він знаходиться у префіксі на першому місці, то замість змінної, яка зв'язана цим квантором, скрізь у матриці поставити деяку сколемівську сталу, яка у формулі ще не зустрічалась, а квантор існування видалити.

2. Якщо квантор існування не на першому місці префікса, наприклад $\forall x_1 \forall x_2 ... \forall x_n \exists x_i$ то замість змінної x_i скрізь у матриці поставити деяку сколемівську функцію $f(x_1 x_2,...x_n)$, яка у формулі ще не зустрічалась, а квантор існування видалити.

3.Перейти до пункту 1, аж поки не видалиться останній квантор існування. У результаті таких перетворень отримується нова формула: $A_s = \forall x_1 \forall x_2 ... \forall x_n (B^*)$, яка називається *сколемівською стандартною формою (ССФ).*

Приклад 3. Розглянемо покроково сколемізацію формули $A = \exists x \forall y \forall z \exists v \forall w \exists t (P(x,y) \vee Q(z,v,w) \wedge R(v,t))$.

1. Замість x вводимо сталу a : $\forall y \forall z \exists v \forall w \exists t (P(a,y) \vee Q(z,v,w) \wedge R(v,t))$;

2. Комбінація кванторів $\forall y \forall z \exists v$ читається «для довільних y та z існує і може трактуватись, як означення деякої функції. Після підстановки цієї функції у матрицю отримаємо нову формулу: $\forall y \forall z \forall w \exists t (P(a,y) \vee Q(z, f(y,z), w) \wedge R(f(y,z),t))$;

3.Комбінація кванторів $\forall y \forall z \forall w \exists t$ рівносильна існуванню функції $f=g(y,z,w)$. Вилучаємо останній квантор існування і маємо ССФ для А: $A_s = \forall y \forall z \forall w (P(a,y) \vee Q(z, f(y,z), w) \wedge R(f(y,z), g(y,z,w)))$.

У ССФ сколемівські функції і сталі вибираються довільно, тому у загальному випадку формули A та A_s не рівносильні. Але при дослідженні типу формули корисне твердження :

Теорема 3.3.5. Формула A є суперечністю тоді і тільки тоді, коли її сколемівська стандартна форма A_s є суперечністю.

Запитання для самоконтролю

1. Яка формула β називається логічним наслідком множини формул $\{\alpha_1, \alpha_2, ..., \alpha_n\}$?
2. Що таке силогізми? Як їх записують за допомогою категоричних суджень?
3. Які дві формули логіки предикатів називаються рівносильними?
4. Які рівносильності називають законами перейменування зв'язаних змінних?
5. Які рівносильності називають законами де Моргана для кванторів?
6. Які рівносильності називають законами пронесення кванторів?
7. Які рівносильності називають законами перейменування зв'язаних змінних?
8. Що називається зведеною формою (ЗФ) для даної формули логіки предикатів?
9. Що називається пренексною або випередженою нормальною формою (ПНФ, ВНФ) для даної формули логіки предикатів?

10. Що називається сколемівською стандартною формою (ССФ) для даної формули логіки предикатів?
11. Які алгоритми побудови рівносильних ЗФ та ПНФ для даної формули логіки предикатів?
12. В чому суть процесу сколемізації формули логіки предикатів?

ВПРАВИ

3.21. Перевірити, чи правильно стоїть знак |= у співвідношеннях;

a) $\forall x \exists y P(x,y) \models \exists x \forall y P(x,y)$;

b) $P(x) \to R \models \exists x P(x) \to R$;

c) $\forall x (P(x) \to R(x)) \vee P(x) \models R(x)$;

d) $\forall x (P(x) \vee B(x)) \models \forall x P(x) \vee \forall x R(x)$.

3.22. Використовуючи поняття логічного наслідку, перевірити, чи будуть логічними такі міркування:

a) Ніхто не може розв'язати задачу не знаючи алгоритму розв'язку. Студент не розв'язав задачу. Отже студент не знає алгоритму розв'язку.

b) Будь-яка людина смертна. Вовк смертний. Отже вовк – людина.

c) Кожен студент за весь термін навчання пропускає хоч одну пару. Викладачі не поважають студентів, які пропускають пари. Отже викладачі не поважають всіх студентів.

d) Не всі алгебраїчні числа є раціональними. Всі раціональні числа – дійсні. Отже, деякі алгебраїчні числа не належать до дійсних чисел.

e) Кожна нескінченна множина має зчисленну підмножину. Множина всіх ірраціональних чисел нескінченна. Кожна множина має потужність не меншу, ніж будь-яка її підмножина. Отже потужність множини ірраціональних чисел не менша ніж зчисленна.

f) Деякі неперервні функції диференційовані в області їхнього означення. Існують тригонометричні функції, які диференційовані в області їхнього означення. Отже, деякі тригонометричні функції неперервні.

g) Кожний математик мислить логічно. Той, хто мислить логічно, не робить логічних помилок. Іван робить логічні помилки. Отже, Іван – не математик.

3.23. Перевірити, чи існують наступні рівносильності:

a) $\forall x P(x,y) \wedge \forall x Q(x,y,z) \equiv \forall x (P(x,y) \wedge Q(x,y,z))$;

b) $\exists x \forall y P(x,y) \to Q(z) \equiv \forall x \exists y (P(x,y) \to Q(z))$;

c) $\forall y P(x,y) \to \exists x \forall z Q(x,y,z) \equiv \exists y \exists x \forall z \big(P(x,y) \to Q(x,y,z)\big)$;

d) $\forall y \exists x P(x,y,z) \equiv \exists y \forall x \overline{P(x,y,z)}$.

3.24. Побудувати зведені і пренексні нормальні форми та сколемівські стандартні форми для наступних формул логіки предикатів:

a) $\big(\forall x P(x,y) \wedge \exists z Q(x,y,z)\big) \wedge R(x)$;

b) $\forall x \big(\exists x \big(P(x,y) \to \forall y R(y)\big) \wedge \forall y \exists z Q(x,y,z)\big)$;

c) $\Big(\overline{\forall x \exists y P(x,y)} \wedge \forall y Q(y,z)\Big) \to P(x,y)$;

d) $\overline{\forall x \big(\exists y P(x,y) \to Q(x)\big)} \vee \exists x R(x)$;

e) $\big(\forall x \exists y P(x,y,z) \to \exists z Q(y,z)\big) \to \forall x R(x)$;

f) $\Big(\forall x P(x) \vee \exists y \big(\overline{P(x) \wedge Q(x,y)}\big)\Big) \leftrightarrow \overline{\exists z R(z)}$.

3.25. Розв’язати наступні рівняння, системи рівнянь та нерівності, попередньо записавши відповідні їм формули:

a) $(x-1)(x+2)(2x-3)=0$;

b) $\begin{cases} (x-1)(x-3)(2x+y)=0, \\ (x+1)(x-2)(x+3y)=0; \end{cases}$

c) $\dfrac{2y-1}{x^2-4}=0$;

d) $\begin{cases} x^2-5x+4=0, \\ y^2-3y+2=0; \end{cases}$

e) $\dfrac{(x+1)(2x+3)}{x^2-9}>0$;

f) $|x-1|>5$;

g) $|x^2-x+6|<8$.

3.4. МЕТОД РЕЗОЛЮЦІЙ

Основна ідея методу резолюцій, який розглядається у логіці висловлень, зберігається і у логіці предикатів. Нагадаємо основні означення і факти.

Бінарною резольвентою $R(D_1, D_2)$ двох диз’юнктів D_1, D_2 називається диз’юнкція літералів, що залишається після видалення пари контрарних.

Наприклад, якщо $D_1 = p$, $D_2 = \overline{p} \vee q$ то $R(D_1, D_2) = q$. Тут резольвента є висновком, який отримується за правилом modus ponens з посилок p та $p \to q$. Аналогічно резольвента $D_1 = \overline{p} \vee q$ і $D_2 = \overline{q} \vee t$ ілюструє правило силогізму: $R(D_1, D_2) = \overline{p} \vee t$. Отже, правило резолюцій є сильнішим з усіх схем логічного висновку. Це випливає з теореми, яка справедлива у самому загальному випадку.

Теорема 3.4.1 Якщо для диз'юнктів D_1, D_2 існує резольвента *R(*D_1, D_2*),* то вона є логічним наслідком цих диз'юнктів: $D_1, D_2 \models R(D_1, D_2)$.

Множина диз'юнктів $D_1, D_2, ..., D_n$ називається *невиконуваною*, якщо формула $F = D_1 \wedge D_2 \wedge ... \wedge D_n$ тотожно хибна.

Якщо можна встановити, що деяка формула *F* хибна, то можна відповісти, чи є логічне слідування $\alpha_1, \alpha_2, ..., \alpha_n \models \beta$, оскільки для цього потрібно дослідити, чи буде формула $F = \alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n \wedge \bar{\beta}$ хибною.

Методом резолюцій називається послідовне отримання бінарних резольвент із заданих диз'юнктів та з усіх тих, що утворюються.

Застосовуючи метод резолюцій, можна отримати резольвенту, у якій не залишиться жодного літерала. Кажуть, що отримали порожній диз'юнкт □.

Теорема 3.4.2 (про повноту методу резолюцій). Множина диз'юнктів S невиконувана тоді і тільки тоді, коли у результаті застосування методу резолюцій до множини S отримується порожній диз'юнкт □.

Є багато різних процедур для реалізації методу резолюцій: локрезолюція, метод насичення рівня, стратегія викреслювання тощо.

У логіці предикатів для дослідження невиконуваності множини диз'юнктів потрібно провести додаткову процедуру уніфікації формул. Тут літералом є елементарна формула, терми якої можуть містити змінні, сталі або вирази із функціональних символів і термів. $P(x, f(y), b)$, $\overline{Q(g(x,a))}$ $S(x_1, x_2, f(x_1, x_2))$ - приклади літералів.

Підстановкою у літерали термів замість змінних можна отримати різні *частинні випадки (приклади)* літерала. Наприклад, частинними випадком першого літерала можуть бути *P(z,f(a),b)*, *P(g(z),f(c),b)*, *P(c,f(a),b)*. Останній частинний випадок називається *атомом*, бо не містить змінних. Підстановку терма *t* замість змінної *x* позначають $\{t/x\}$. Одночасно можна виконати кілька замін. їх групують у підстановку. Наприклад, перший частинний випадок отримано у результаті підстановки $\theta_1 = \{c/x, a/y\}$ другий - $\theta_2 = \{g(z)/x, c/y\}$, третій $\theta_3 = \{c/x, a/y\}$. У загальному випадку $\theta = \{t_1/x_1, t_2/x_2 ... t_k/x_k\}$ де всі x_k - різні змінні, t_k - терми. Застосування підстановки θ до літерала позначають P_0. Послідовне виконання двох підстановок θ_1 та θ_2 дає третю

$\theta_3 = \theta_1 \circ \theta_2$.

Множина літералів $\{L_1, L_2, \dots L_n\}$ називається *уніфікованою,* якщо існує така підстановка θ, що $(L_1)_\theta = (L_2)_\theta = ... = (L_n)_\theta$. Підстановка θ називається *уніфікатором* множини літералів $\{L_i\}$. Уніфікатор σ множини формул називається *найзагальнішим*, якщо для кожного уніфікатора θ цієї множини існує підстановка λ, що $\theta = \sigma \circ \lambda$.

Існує алгоритм уніфікації, який починає роботу з порожньої підстановки і крок за кроком знаходить множину неузгодженості в літералах і будує найзагальніший уніфікатор, якщо він є. Наприклад, для літералів $L_1 = P(x, f(y), b)$ та $L_1 = P(a, f(b), b)$ перша множина неузгодженості *W1={x,a)*. Щоб ліквідувати неузгодженість, робимо підстановку $\theta_1 = \{a/x\} : (L_1)_{\theta_1} = P(a, f(y), b)$. Друга множина неузгодженості *W2={y,b}*. Після підстановки $\theta_2 = \{b/y\}$ *у* новоутворений літерал $((L_1)_{\theta_1})_{\theta_2} = P(a, f(b), b)$ отримаємо однакові літерали. Отже, $\theta = \theta_1 \circ \theta_2$ є найзагальнішим уніфікатором. Кожен елемент множини неузгодженості повинен бути термом або літералом. Якщо множина неузгодженості не містить змінних, то така множина літералів не уніфікується.

Нехай деякий диз'юнкт D містить літерали $\{L_1, L_2, ..., L_K\}$для яких існує спільний уніфікатор. Тоді замість к літералів залишають один і така процедура називається *склейкою*.

Приклад 1. Знайти найзагальніший уніфікатор, якщо множина літералів уніфікується:

a) $\{P(a), P(f(y))\}$;

b) $\{P(x), P(f(a))\}$;

c) $\{P(a, f(x)), P(x, y)\}$.

Розв'язання. а) Для нескладних літералів алгоритм уніфікації очевидний. Порівнюючи у літералах символи зліва направо, отримуємо множину неузгодженості $W_1 = \{a_1, f(y)\}$. Вона не містить змінних, тому не можна здійснити жодної підстановки і множина літералів не уніфікується.

b) Множина неузгодженості $W_1 = \{x, f(a)\}$. Вона містить змінну x, тому можлива підстановка $\theta_1 = \{f(a)/x\}$ і множина літералів набуде вигляду $\{P(f(a)), P(f(a))\}$. Це однакові літерали, тому $\theta_1 = \{f(a)/x\}$ є найзагальнішим уніфікатором.

с) У більш складних випадках бажано діяти за алгоритмом:

1. Вводимо порожню підстановку ε. Початкова множина диз'юнктів $S_0 = \{P(a, f(x), P(x, y)\}$ не одиничний диз'юнкт, тому ще не отримали найзагальніший уніфікатор;

2. Для S_0 множина неузгодженості $W_0 = \{a, x\}$. Робимо підстановку $\theta_1 = \varepsilon \circ \{a / x\} : S_1 = (S_0)_{\theta_1} = \{P(a, f(a), P(a, y)\}$ - не одиничний диз'юнкт;

3. Для S_1 множина неузгодженості $W_1 = \{f(a), y\}$. Робимо підстановку $\theta_2 = \theta_1 \circ \{f(a) / y\}$: $S_2 = (S_0)_{\theta_2} = \{P(a, f(a)), P(a, f(a))\} = \{P(a, f(a))\}$. Множина диз'юнктів перетворилась на один диз'юнкт, тому $\theta_2 = \{a / x, f(a) / y\}$- найзагальніший уніфікатор.

Нехай є два диз'юнкти D_1 і D_2 і всі змінні у них різні. Диз'юнкт D_1, містить літерал L_1, а диз'юнкт D_2 містить літерал L_2, причому існує уніфікатор θ такий, що $(L_1)_\theta = (\overline{L_2})_\theta$. *Бінарною резольвентою* диз'юнктів D_1 і D_2 у логіці предикатів називається диз'юнкція літералів, які залишаються після викреслювання контрарних уніфікованих.

Перед побудовою резольвент спочатку виконують склейку у кожному диз'юнкті (якщо це можливо). У логіці предикатів теж справедлива теорема про повноту методу резолюцій. Наприклад, для перевірки логічного слідування $\alpha_1, \alpha_2, ..., \alpha_n \models \beta$ методом резолюцій потрібно:

- Утворити формулу $F = \alpha_1 \wedge \alpha_2 \wedge ... \wedge \alpha_n \wedge \bar{\beta}$. При цьому у кожній посилці та висновку змінні перейменувати, позначивши усі різними буквами, оскільки вони можуть мати різний зміст;
- Знайти сколемівську стандартну форму цієї формули $F_s = \forall x_1 \forall x_2 ... \forall x_n (F^*)$ і записати її матрицю у кон'юнктивній нормальній формі $F^* = D_1 \wedge D_2 \wedge ... D_k$;
- До множини диз'юнктів $D_1, D_2 ... D_n$ застосувати метод резолюцій провівши, за необхідності, уніфікацію літералів. Якщо у результаті побудови всіх можливих резольвент отримується порожній диз'юнкт □, то множина диз'юнктів невиконувана, формули F_S і F тотожно хибні, отже, є логічне слідування.

Приклад 2. Перевірити за допомогою методу резолюцій, чи правильні наступні схеми логічних слідувань:

a) $\forall x(M(x) \to \overline{P(x)})$, $\forall x(S(x) \to M(x)) \vDash \forall x(S(x) \to \overline{P(x)})$;

b) $\forall x(P(x) \to S(x))$, $\forall x(Q(x) \to \overline{P(x)}) \vDash \forall x(Q(x) \to \overline{S(x)})$.

Розв'язання. а) З посилок та висновку утворимо формулу, яку потрібно перевірити на не виконуваність. Змінні у формулах в областях дії кванторів перейменуємо по різному, оскільки вони можуть мати різний зміст : $F = \forall x(M(x) \to \overline{P(x)}) \wedge \forall y(S(y) \to M(y)) \wedge \overline{\forall z(S(z) \to \overline{P(x)})}$. Зведемо формулу до випередженої нормальної форми:

$$F \equiv \forall x(\overline{M(x)} \vee \overline{P(x)}) \wedge \forall y(\overline{S(y)} \vee M(y)) \wedge \exists z(\overline{\overline{S(z)} \vee \overline{P(z)}}) \equiv \forall x(\overline{M(x)} \vee \overline{P(x)}) \wedge \forall y(\overline{S(y)} \vee$$
$$\vee M(y)) \wedge \exists z(S(z) \wedge P(z)) \equiv \exists z \forall x \forall y((\overline{M(x)} \vee \overline{P(x)}) \wedge (\overline{S(y)} \vee M(y)) \wedge S(z) \wedge P(z)).$$

Квантор існування можна винести першим, оскільки для кон'юнкції справедливий переставний закон. Це спростить запис сколемівської форми:

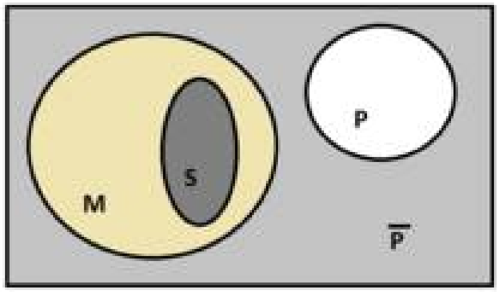
$F_s = \forall x \forall y((\overline{M(x)} \vee \overline{P(x)}) \wedge (\overline{S(y)} \vee M(y)) \wedge S(a) \wedge P(a))$. Випишемо множину диз'юнктів:

(1) $\overline{M(x)} \vee \overline{P(x)}$;

(2) $\overline{S(y)} \vee M(y)$;

(3) $S(a)$;

(4) $P(a)$.



а)

Далі проводимо уніфікацію літералів та утворюємо резольвенти, виписуючи їх за методом насичення рівня:

(5) $\overline{P(x)} \vee \overline{S(x)}$ із (1), (1) після $\{x / y\}$;

(6) $\overline{M(a)}$ із (1), (4) після $\{a / x\}$;

(7) $M(a)$ із (2), (3) після $\{a / y\}$;

(8) □ із (6) і (7).

Оскільки отримали порожній диз'юнкт, то формула невиконувана, тому дане логічне слідування є правильним. Це один з правильних силогізмів Арістотеля *Celarent.* З ілюстрації видно, що з даних посилок висновок є очевидним.

b) Записуємо потрібну формулу та зводимо її до сколеміської форми:

$$F \equiv \forall x(P(x) \to S(x)) \wedge \forall y(Q(y) \to \overline{P(y)}) \wedge \overline{\forall z(Q(z) \to \overline{S(z)})} \equiv$$
$$\equiv \forall x(\overline{P(x)} \vee S(x)) \wedge \forall y(\overline{Q(y)} \vee \overline{P(y)}) \wedge \exists z(Q(z) \wedge S(z));$$
$$F_x \equiv \forall x \forall y((\overline{P(x)} \vee S(x)) \wedge \forall y(\overline{Q(y)} \vee \overline{P(y)}) \wedge Q(a) \wedge S(a).$$

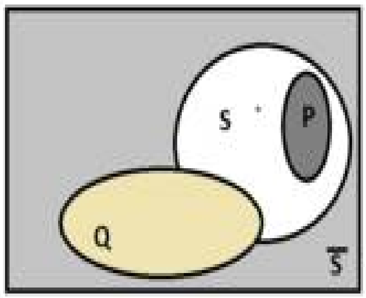
Випишемо множину диз’юнктів:

(1) $\overline{P(x)} \vee S(x)$;

(2) $\overline{Q(y)} \vee \overline{P(y)}$;

(3) $Q(a)$;

(4) $S(a)$.



Утворюємо резольвенти: (5) $\overline{P(a)}$ із (2),(3) після $\{a/y\}$. b)

Ніяких інших резольвент утворити не можна. Отже вивід порожнього диз’юнкта не існує. За теоремою про повноту методу резолюцій можна стверджувати, що висновок не є логічним наслідком даних посилок. З ілюстрації видно, що при правильності всіх посилок висновок може не виконуватись : не всі Q належать $\overline{S}$. Більше того, можлива ситуація, коли всі Q належать S.

Приклад 3. Чи правильні міркування:

a) Всі люди смертні. Сократ – людина. Отже він смертний;

b) Всі, хто вміє обчислювати інтеграли, математики. Діти – не математики. Деякі діти мають математичні здібності. Отже, дехто з тих, хто має математичні здібності, не обчислює інтегралів.

Розв’язання. a) Введемо предикати L(x)= «*x – людина* », C(x)= «*x – смертний*» та елемент c=*Сократ*. Логічне слідування можна записати :

$\forall x(L(x) \to C(x)), L(c) \vDash C(c)$. Далі: $F = \forall x(L(x) \to C(x)) \wedge L(c) \wedge \overline{C(c)} \equiv$

$\equiv \forall x((\overline{L(x)} \vee C(x)) \wedge L(c) \wedge \overline{C(c)}) = F_s$. Виписуємо диз’юнкти і резольвенти :

(1) $\overline{L(x)} \vee C(x)$;

(2) $L(c)$;

(3) $\overline{C(c)}$.

(4) C(c) із (1), (2) після $\{c/x\}$;

(5) $\overline{L(c)}$ із (1), (3) після $\{c/x\}$;

(6) □ із (2), (6).

Отже, міркування логічне і правильне.

c) Введемо предикати *I(x) = «x – вміє обчислювати інтеграли», M(x) = «x - математик», З(x) = «x – має математичні здібності», D(x) =«x - дитина».* Тоді всі посилки та висновок можна записати :

$\forall x(I(x) \to M(x)), \forall x(D(x) \to \overline{M(x)}), \exists x(D(x) \wedge З(x)) \vDash \exists x(З(x) \wedge \overline{I(x)})$. Далі :

$$F=\forall x(I(x)\to M(x))\wedge\forall y(D(y)\to\overline{M(y)})\wedge\exists z(D(z)\wedge З(z))\wedge\overline{\exists t(З(t)\wedge\overline{I(t)})}\equiv$$
$$\forall x(\overline{I(x)}\vee M(x))\wedge\forall y(\overline{D(y)}\vee\overline{M(y)})\wedge\exists z(D(z)\wedge З(z))\wedge\forall t(\overline{З(t)}\vee I(t))\equiv$$
$$\exists z\forall x\forall y\forall t((\overline{I(x)}\vee M(x))\wedge(\overline{D(y)}\vee\overline{M(y)})\wedge(D(z)\wedge З(z))\wedge(\overline{З(t)}\vee I(t))).$$
$$F_s=\forall x\forall y\forall t((\overline{I(x)}\vee M(x))\wedge(\overline{D(y)}\vee\overline{M(y)})\wedge(\overline{З(t)}\vee I(t))\wedge D(a)\wedge З(a)).$$

Виписуємо диз’юнкти і резольвенти :

(1) $\overline{I(x)}\vee M(x)$;

(2) $\overline{D(y)}\vee\overline{M(y)}$;

(3) $\overline{З(t)}\vee I(t)$;

(4) $D(a)$;

(5) $З(a)$.

(6) $\overline{I(y)}\vee\overline{D(y)}$ із (1), (2) після $\{y/x\}$;

(7) $\overline{З(t)}\vee M(t)$ із (1), (3) після $\{t/x\}$;

(8) $\overline{M(a)}$ із (2), (4) після $\{a/y\}$;

(9) $I(a)$ із (3), (5) після $\{a/y\}$;

(10) $\overline{I(a)}$ із (1), (8) після $\{a/x\}$.

(11) □ із (9), (10).

Тут не виписані зайві диз’юнкти за методом насичення рівня, оскільки порожній диз’юнкт отримується очевидно. Отже, дане міркування логічне і правильне.

<u>Запитання для самоконтролю</u>

1. Що називається бінарною резольвентою двох диз’юнктів у логіці предикатів?
2. Який процес називається методом резолюцій?
3. Що таке порожній диз’юнкт?
4. В чому суть процедури уніфікації формул логіки предикатів?
5. Яка множина літералів $\{L_1, L_2, \ldots L_n\}$ називається уніфікованою?
6. Що таке множина неузгодженості літералів? Наведіть приклад.
7. Що таке найзагальніший уніфікатор?
8. Яка процедура називається склейкою?
9. Яка множина диз’юнктів називається невиконуваною?
10. В чому суть теореми про повноту резолюцій?
11. За якими кроками виконують перевірку на загальнозначущість

(тавтологію) формули логіки предикатів методом резолюцій?

12. Як перевірити логічність міркувань за допомогою методу резолюцій?

ВПРАВИ

3.26. Знайти найзагальніший уніфікатор, якщо множина літералів уніфікується:

a) $\{P(a), P(f(b)\}$;

b) $\{P(x), P(f(x))\}$;

c) $\{P(y, f(x)), P(a, z)\}$;

d) $\{P(a, x, f(g(y))), P(z, f(z), f(u))\}$;

e) $\{Q(f(a), g(x)), Q(y, y)\}$.

3.27. Перевірити за допомогою методу резолюцій, чи правильні наступні схеми логічних слідувань:

a) $\forall x(M(x) \to \overline{P(x)})$, $\exists x(S(x) \wedge M(x)) \vDash \exists x(S(x) \wedge \overline{P(x)})$;

b) $\forall x(P(x) \to \overline{S(x)}),\ \ \forall x(Q(x) \to \overline{P(x)}) \vDash \exists x(Q(x) \wedge \overline{S(x)})$;

c) $\forall x(M(x) \to P(x)),\ \ \forall x(M(x) \to S(x)) \vDash \exists x(S(x) \wedge P(x))$;

d) $\forall x(P(x) \to \overline{M(x)}), \exists x(S(x) \wedge M(x)) \vDash \exists x(S(x) \wedge \overline{P(x)})$;

e) $\exists x(P(x) \wedge Q(x)), \forall x(P(x) \to R(x) \wedge T(x)) \vDash \exists x(Q(x) \wedge T(x))$;

f) $\exists x(P(x) \wedge \forall y(Q(x) \to R(x, y))), \forall x(P(x) \to \forall y(S(y) \to \overline{R(x, y)})) \vDash \forall x(Q(x) \to \overline{S(x)})$;

g) $\forall y(S(y) \to C(y)) \vDash \forall x(\exists y S(y) \wedge V(x, y)) \to \exists z(C(z) \wedge V(x, z))$.

3.28. Довести правильність силогізмів Арістотеля. Навести приклад міркувань за кожною схемою:

a) $\forall x(M(x) \to P(x)), \exists x(S(x) \wedge M(x)) \vDash \exists x(S(x) \wedge P(x))$ -фігура I, Darii;

b) $\forall x(M(x) \to \overline{P(x)}), \exists x(S(x) \wedge M(x)) \vDash \exists x(S(x) \wedge \overline{P(x)})$ -фігура I, Ferio;

c) $\forall x(M(x) \to P(x)), \exists x(S(x) \wedge M(x)) \vDash \exists x(S(x) \wedge \overline{P(x)})$ -фігура II, Baroko;

d) $\forall x(M(x) \to \overline{P(x)}), \exists x(M(x) \wedge S(x)) \vDash \exists x(S(x) \wedge \overline{P(x)})$ -фігура III, Ferison;

e) $\exists x(P(x) \wedge M(x)), \forall x(M(x) \to S(x)) \vDash \exists x(S(x) \wedge P(x))$ -фігура IV, Dimaris.

3.29. Перевірити, чи правильні міркування на базі логіки предикатів:

a) всі цілі числа – раціональні. Деякі дійсні числа – цілі. Отже, деякі дійсні числа – раціональні;

b) деякі парні функції періодичні. Жодна монотонна функція не є парною. Отже, жодна монотонна функція не періодична;

c) всі студенти КДПУ живуть у Кіровоградській області. Деякі жителі Кіровоградської області – пенсіонери. Отже, деякі студенти КДПУ – пенсіонери;

d) всі вчителі – педагоги. Деякі педагоги нагороджені орденом. Отже, деякі вчителі нагороджені орденом;

e) кожний атлет сильний. Кожний, хто сильний і розумний, доб’ється успіху. Петро атлет і розумний. Отже, він доб’ється успіху;

f) всі боксери – спортсмени. Жоден спортсмен не курить. Отже, жоден курець не боксер;

g) декому подобається співати. Дехто не любить нікого з тих, хто любить співати. Отже, декого люблять не всі;

h) деякі студенти старанні. Жоден студент не позбавлений здібностей. Отже, деякі студенти, позбавлені здібностей, не старанні;

i) дурень був би здатен на таке. Я не здатен. Отже, я не дурень;

j) кожний, хто може розв’язати цю задачу – математик. Жоден математик не може її розв’язати. Отже, задача не має розв’язку;

k) кожний радикал є прибічником прогресу. Деякі консерватори недолюблюють всіх прибічників прогресу. Отже, деякі консерватори недолюблюють всіх радикалів;

l) формула логіки висловлень є тавтологією, якщо всі її диз’юнктивні одночлени містять контрарні пари літералів. Задана формула не тавтологія. Отже, принаймні один її диз’юнктивний одночлен не містить контрарної пари літералів;

m) Деякі змії ядовиті. Ужі – змії. Отже, ужі ядовиті.

3.30. Перевірити за допомогою методу резолюцій, чи є наступні формули загальнозначущими (тавтологіями логіки предикатів):

a) $\exists x P(x) \to \overline{\forall x \overline{P(x)}}$;

b) $\exists x \forall y R(x,y) \to \forall y \exists x R(x,y)$;

c) $\forall x [P(x,x) \to (R(x) \to \forall x(\forall x P(x,x) \wedge R(x)))]$;

d) $\exists x \exists y (P(x,y) \to R(x)) \to \forall x(\overline{\exists y P(x,y)} \vee R(x))$.

3.5. ЗАСТОСУВАННЯ ЛОГІКИ ПРЕДИКАТІВ

3.5.1 Застосування символіки логіки предикатів в математичних формулюваннях

Під час вивчення математичних дисциплін різного спрямування доволі широко використовують символіку математичної логіки для скорочення запису того чи іншого математичного твердження. Особливо це стосується символіки логіки предикатів.

Досить часто в математичних формулюваннях предметні змінні під знаком кванторів пробігають певну підмножину множини задання предикатів. Для прикладу розглянемо наступну формулу $\forall x \exists y P_1^2(f_1^1(x), y)$. Множиною інтерпретації оберемо множину дійсних чисел, функціональному символу $f_1^1(x)$ поставимо у відповідність операцію відшукання квадратного кореня, а предикатному символ $P_1^2(x, y)$ – предикат «$x = y$». Відмітимо, що наш предикат заданий на множині R^2, але формула $\forall x \exists y \left(\sqrt{x} = y\right)$ істинна лише для невід'ємних значень x та y. Тому, зі змістовної математичної точки зору це запишеться так $\forall x \in R^+ \exists y \in R^+ (y = \sqrt{x})$.

Також, при вивченні математичних дисциплін дуже часто зустрічаються вирази «існує натуральне число», «для будь –якого дійсного числа», «існує єдиний елемент х» і т. д. . Тому в математиці використовуються так звані *обмежені квантори*. Це по суті скорочені записи тих чи інших формул логіки предикатів, а точніше – їх інтерпретацій. Наприклад, формули $\forall x (x \in M \to P(x))$ і $\exists x (x \in M \to P(x))$ з використанням обмежених кванторів загальності та існування запишуться наступним чином:

$$\forall_{x \in M} x (P(x)) \text{ і } \exists_{x \in M} x (P(x)).$$

В подальшому будемо використовувати дещо зручнішу форму запису, а саме:

$$\forall x \in M (P(x)) \text{ і } \exists x \in M (P(x)).$$

Тобто, для формули з нашого прикладу в тій інтерпретації ми мали б записати наступний вираз без використання обмежених кванторів:

$$\forall x \left((x \in R^+) \to \exists y \left((y \in R^+) \to (y = \sqrt{x})\right)\right).$$

Також, при записі математичних понять використовують так званий квантор існування та єдиності $\exists! x$, який читається: «існує єдиний елемент x, такий, що…». Наприклад, запис $\exists! x P(x)$ в розгорнутому вигляді є формулою

$$\exists x P(x) \wedge \forall x \forall y \left(P(x) \wedge P(y) \to x = y\right).$$

3.5.2 Використання елементів математичної логіки для аналізу структури математичних теорем

Логіка предикатів дозволяє проаналізувати будову математичних теорем. В математичній науці найчастіше зустрічаються теореми 4 типів, які називають категоричними судженнями і які ми розглядали у пункті 3.1. Їх класифікація має наступний вигляд:

A: «Всі *S* суть *P*» – загальноствердживальне судження;

E: «Будь-яке *S* не є *P*» – загальнозаперечувальне судження;

I: «Деякі *S* суть *P*» – частковоствердживальне судження;

O: «Деякі *S* не є *P*» – частковозаперечувальне судження.

В термінах логіки предикатів судження відповідних типів записуються наступним чином.

A: $\forall x(S(x) \rightarrow P(x))$. Тобто судження типу A розуміють так: для будь-якого елемента x , якщо він має властивість *S(x)*, то він має і властивість *P(x)*. Прикладом такого судження може бути наступне: якщо функція диференційована, то вона неперервна.

E: $\forall x\left(S(x) \rightarrow \overline{P(x)}\right)$. На мові логіки предикатів це означає наступне: для будь-якого елемента x , якщо він має властивість *S(x)*, то він не має властивості *P(x)*. Прикладом може бути судження: Якщо многокутник є трикутником, то він не може бути ромбом.

I: $\exists x(S(x) \wedge P(x))$. В межах логіки предикатів судження типу I потрібно розуміти так: існує елемент *x*, який має властивість *S(x)* і одночасно має властивість *P(x)*. Приклад: існують такі функції є неперервними і не диференційованими одночасно.

O: $\exists x\left(S(x) \wedge \overline{P(x)}\right)$. З позиції логіки предикатів це судження потрібно розуміти так: існує елемент *x*, який має властивість *S(x)* і одночасно не має властивості *P(x)*. Як приклад можна взяти наступне судження: існують такі многокутники, які є опуклими і не є чотирикутниками.

В математичній теорії особливий інтерес становлять судження типу A, які можна записати наступним чином, використовуючи обмежені квантори:

$$\forall x \in M(S(x) \rightarrow P(x)),$$

де *S(x)* і *P(x)* – деякі предикати, задані на множині *M*, які описують ті чи інші властивості елементів цієї множини. Така конструкція теореми відповідає в математиці конструкції «якщо…, то…» або «нехай…, тоді…». В наведеному формулюванні чітко виділяють умову теореми *S(x)* та висновок теореми *P(x)*.

Теорема вважається вірною, якщо множина істинності M_S предиката *S(x)* непорожня і є підмножиною множини істинності предиката *P(x)*. При цьому *S(x)* називають *достатньою умовою* для *P(x)*, а *P(x)* – *необхідною умовою* для *S(x)*.

Також, як відомо ще з курсу шкільної математики, часто виділяють обернену та протилежну теореми. Так за наявності теореми

$$\forall x \in M\big(S(x) \to P(x)\big)$$

твердження

$$\forall x \in M\big(P(x) \to S(x)\big)$$

називається *оберненим* для даної теореми, твердження

$$\forall x \in M\big(\overline{S(x)} \to \overline{P(x)}\big)$$

протилежною теоремою до теореми $\forall x \in M\big(S(x) \to P(x)\big)$, а твердження

$$\forall x \in M\big(\overline{P(x)} \to \overline{S(x)}\big)$$

протилежним до оберненої теореми $\forall x \in M\big(P(x) \to S(x)\big)$.

Розглянемо спочатку поняття оберненої теореми. Отже теорема $\forall x \in M\big(S(x) \to P(x)\big)$ – пряма, а теорема $\forall x \in M\big(P(x) \to S(x)\big)$ – обернена до неї. Тоді для них можуть бути справедливими чотири випадки.

1) Обидві теореми вірні. Прикладом може бути твердження «Якщо у чотирикутника протилежні сторони взаємно паралельні, то такий чотирикутник паралелограм», і обернена їй «Якщо чотирикутник є паралелограмом, то його протилежні сторони – паралельні».

2) Пряма теорема вірна, а обернена – ні. Це можна побачити на такому прикладі: пряма теорема «Якщо функція інтегрована за Ріманом, то вона інтегрована і за Лебегом» є вірною, а обернена – «Якщо функція інтегрована за Лебегом, то вона інтегрована і за Ріманом» – невірна.

3) Пряма теорема невірна, а обернена вірна. Для цього випадку можна взяти в попередньому прикладі обернену теорему за пряму, а пряму – за обернену.

4) Обидві теореми невірні. Як приклад розглянемо твердження «Якщо число ділиться на 2, то воно ділиться і на 3» – пряма теорема, яка є не вірною, і обернена – «Якщо число ділиться на 3, то воно ділиться і на 2» – також є невірною.

Розглянемо окремо випадок, коли обидві теореми, і пряма, і обернена, є вірними. Тоді кожне з них є одночасно і необхідною, і достатньою умовою для іншої. В такому випадку, для запису теорем використовують одну формулу

$$\forall x\left(S(x) \leftrightarrow P(x)\right),$$

і формулювання обох теорем читають «для кожного елемента $x \in M$ $S(x)$ має місце тоді і тільки тоді, коли має місце *P(x)*». Прикладом такого твердження може бути наступне: натуральне число ділиться на 6 тоді і тільки тоді, коли воно одночасно ділиться і на 2, і на 3.

Далі розглянемо теорему $\forall x \in M\left(\overline{S(x)} \to \overline{P(x)}\right)$, яка є протилежною до теореми $\forall x \in M\left(S(x) \to P(x)\right)$. Такі теореми називаються взаємно протилежними. Також взаємно протилежними є теореми $\forall x \in M\left(P(x) \to S(x)\right)$ і $\forall x \in M\left(\overline{P(x)} \to \overline{S(x)}\right)$. Між цими теоремами існує досить тісний зв'язок, а саме:

$$\forall x \in M\left(S(x) \to P(x)\right) \equiv \forall x \in M\left(\overline{P(x)} \to \overline{S(x)}\right),$$

$$\forall x \in M\left(P(x) \to S(x)\right) \equiv \forall x \in M\left(\overline{S(x)} \to \overline{P(x)}\right).$$

Ці рівносильності використовуються в математиці при так званому непрямому доведенні або доведенні від супротивного, коли, наприклад, замість теореми $\forall x \in M\left(S(x) \to P(x)\right)$ легше довести теорему $\forall x \in M\left(\overline{P(x)} \to \overline{S(x)}\right)$.

Запитання для самоконтролю

1. Що розуміють під терміном обмежені квантори та як вони використовуються.
2. Які формули логіки предикатів визначають обмежені квантори.
3. Якою формулою логіки предикатів запишеться твердження: «Існує один і тільки один *x*, що має властивість Р».
4. Якою формулою логіки предикатів запишеться твердження: «Існує не більше однного *x*, що має властивість Р».
5. Що таке категоричні судження та як вони класифікуються.

6. Якою формулами логіки предикатів запишуться: загальноствердживальне судження, частковозаперечувальне судження, частковостверджувальне судження, загальнозаперечувальне судження.
7. При аналізі теореми типу $\forall x\left(S(x) \to P(x)\right)$ що називається достатньою умовою, а що необхідною.
8. Яка теорема називається оберненою до теореми $\forall x\left(S(x) \to P(x)\right)$. Наведіть приклад відповідних математичних тверджень.
9. Яка теорема називається протилежною до теореми $\forall x\left(S(x) \to P(x)\right)$. Наведіть приклад відповідних математичних тверджень.

10. Яка теорема називається протилежною до оберненої. Наведіть приклад прямої та протилежної до оберненої математичних теорем.

<u>ВПРАВИ.</u>

3.31. Записати наступні означення за допомогою мови логіки предикатів:

a) границі функції;
b) збіжного ряду;
c) перпендикулярних прямих;
d) неперервної функції;
e) колінеарних векторів;
f) суми векторів в координатах;
g) паралелограма;
h) рівностороннього трикутника;
i) складеного числа;
j) НСД двох чисел.

3.32. Записати наведені нижче математичні твердження за допомогою мови логіки предикатів та побудувати для кожного з них обернене твердження, протилежне твердження та обернене до протилежного. Визначити, які з них будуть вірними, а які – ні.

a) Якщо діагоналі чотирикутника перетинаються і в точці перетину діляться пополам, то цей чотирикутник паралелограм.

b) Якщо функція неперервна в кожній точці відрізка, то вона неперервна і на всьому відрізку.

c) Для того, щоб ряд був збіжний необхідно щоб його загальний доданок прямував до нуля.

d) Якщо функція неперервна на відрізку і приймає на кінцях цього відрізка значення різних знаків, то на цьому відрізку міститься принаймі один корінь рівняння $f(x)=0$.

e) Якщо дві прямі паралельні деякій третій прямій, то вони паралельні між собою.

3.33. Виділити умову та висновок теореми та сформулювати теореми за допомогою зв’язки «якщо…,то…»:

a). Для подільності добутку двох натуральних чисел на деяке число достатньо подільності на це число одного із співмножників.

b). Рівність трикутників є достатньою умовою їх рівно великості.

c). Необхідною умовою збіжності ряду є те, що його загальний доданок повинен прямувати до 0.

d). На 3 діляться ті числа, у яких сума цифр ділиться на 3.

e). Для того, щоб послідовність була збіжною, необхідно, щоб вона була обмеженою.

4. ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

4.1. ПОНЯТТЯ АЛГОРИТМУ.

Змістовне поняття алгоритму належить до основних понять сучасної математики та інформатики. Це поняття поряд з іншими поняттями (множина, конструктивний об'єкт тощо) не означається через простіші поняття, а описується на прикладах.

Під *алгоритмом* у математиці на змістовному (інтуїтивному) рівні розуміють скінченну впорядковану послідовність точно визначених дій (інструкцій, операцій, команд), виконання яких у вказаному порядку дає можливість розв'язати будь-яку конкретну задачу із деякого даного класу задач. Термін "алгоритм" походить від імені великого середньоазіатського вченого IX століття ал-Хорезмі, який зробив значний внесок у розвиток різних методів обчислень.

Значна частина змісту шкільної математичної освіти так чи інакше пов'язана з оволодінням учнями різними алгоритмами. Вузівський курс математики значно розширює алгоритмічні можливості самої математики та її практичного застосування. Не тільки в математиці, а і в інших науках та різних сферах людської діяльності алгоритмічні процеси займають значне, а в багатьох випадках і вирішальне місце.

Уважний аналіз відомих алгоритмів дає можливість вказати такі основні загальні характерні риси (властивості) алгоритмів:

1). *Фінітність алгоритму* – алгоритм є скінченним об’єктом, що є необхідною умовою його реалізації на машині.

2). *Дискретність алгоритму* – алгоритмічний процес реалізується послідовним виконанням кроків, по одному за одиницю часу (алгоритм виконується в дискретному часі).

3). *Масовість алгоритму* – кожен алгоритм призначений не для виконання одного завдання, а для певного класу однотипних задач.

4). *Елементарність алгоритму* - кожна елементарна операція (такт) алгоритму має формальний характер і може виконуватися автоматично (обчислювачем чи машиною) без урахування її змісту.

5). *Результативність алгоритму*, тобто скінченність процесу перетворення вхідних даних. Застосування алгоритму до конкретного набору вхідних даних завжди завершується одержанням певного результату - розв'язку задачі.

За допомогою алгоритму (якщо він реалізований коректно) кожен конкретний результат отримується за певну скінченну кількість кроків для початкових даних, взятих з певної множини. Тоді кажуть, що для таких початкових даних алгоритм застосовний. Якщо, для якихось початкових даних алгоритм працює необмежено, до кажуть, що до такого набору початкових даних він незастосовний.

Відкриття алгоритму, як самостійного поняття не можна змішувати з відкриттям формальних моделей алгоритму. Такі моделі утворені для формального уточнення інтуїтивного поняття алгоритму. До таких формальних моделей відносяться: машина Тьюрінга, нормальні алгоритми Маркова, машини з натурально значними регістрами, машини Поста та ін.

Іншим підходом до формального уточнення алгоритму є опис певних класів функцій, для яких існує алгоритм знаходження функції за значеннями її аргументів (рекурсивні функції та ін.).

4.2. НОРМАЛЬНІ АЛГОРИТМИ

Розглянемо алгоритмічну систему, що задає клас математично точно визначених алгоритмів, які називаються *нормальними алгоритмами*. Таке уточнення змістовного поняття алгоритму було запропоноване російським математиком А.А. Марковим на початку 50-х років XX ст.

Нормальні алгоритми - це окремий клас алфавітних операторів, які задаються конструктивно, а значить нормальний алгоритм переробляє вхідне слово в деякому алфавіті у вихідне слово в тому ж самому чи іншому алфавіті.

В нормальних алгоритмах використовується тільки один тип операторів дії - *оператори підстановки*, і один тип операторів розпізнавання - *оператори розпізнавання входження одного слова в інше слово*. Для детального опису цих операторів розглянемо ряд понять.

Нехай α і β два слова в деякому алфавіті. Кажуть, що слово β входить в слово α, якщо слово α можна представити у вигляді $\alpha = \alpha_1\beta\alpha_2$ де α_1 і α_2 - деякі слова (можливо і пусті). Знайдене входження слова β в слово α називається *першим зліва* (або просто першим) входженням, якщо в представленні слова α у вигляді $\alpha = \alpha_1\beta\alpha_2$, слово α_1 має найменшу можливу довжину серед всіх подібних представлень слова α.

Розпізнавач входження задається фіксацією деякого слова α, а зміст його застосування полягає в тому, що для будь-якого заданого слова α_1 перевіряється умова - входить чи ні слово α_1 в слово α.

Оператор підстановки задається виразом виду: $\alpha_1 \to \beta_1$. Робота його полягає в тому, що він здійснює підстановку слова β_1 замість першого зліва входження α_1 в задане слово α. Наприклад, якщо α_1 є першим зліва входженням в слово $\alpha = p\alpha_1 q$, то після застосування оператора підстановки $\alpha_1 \to \beta_1$ слово α перетворюється в слово $\gamma = p\beta_1 q$. Домовимось перше зліва входження виділяти за допомогою круглих дужок. Тоді, наприклад, перше входження слова $\alpha_1 = xy$ в слово $\alpha = xxxyyxxyy$ виділимо так: $\alpha = xx(xy)yxxyy$.

Якщо ліва частина підстановки $\alpha_1 \to \beta_1$, тобто слово α_1 входить у задане слово α, то кажуть, що *підстановка* $\alpha_1 \to \beta_1$ *застосовна до слова* α. Якщо ж ліва частина підстановки $\alpha_1 \to \beta_1$ не входить у задане слово α, то кажуть, що *підстановка* $\alpha_1 \to \beta_1$ *незастосовна до слова* α.

Зауважимо, що в нормальних алгоритмах крім звичайних підстановок виду $\alpha_1 \to \beta_1$ можуть використовуватися і так звані *заключні підстановки*, які записуються у вигляді $\alpha_1 \to \bullet\beta_1$ (замість стрілки пишемо стрілку з крапкою). Після виконання будь-якої заключної підстановки дія алгоритму припиняється.

Нормальний алгоритм - це алгоритм, що задається впорядкованою скінченною сукупністю підстановок, які застосовуються до вхідного слова у відповідності з такими правилами:

1). Перевірку застосовності підстановок до перетворюваного слова на будь-якому етапі його переробки треба починати з першої підстановки.

2). Якщо вона застосовна до слова, то після її виконання знову переходимо до п.1).

3). Якщо жодна з перших k підстановок незастосовна до слова, а (k+1)-ша застосовна, то після її виконання знову переходимо до п.1).

4). Процес перетворення вхідного слова продовжується доти, поки не дістанемо слово, до якого жодна з підстановок незастосовна або поки до слова не буде застосовна одна з заключних підстановок.

Сукупність підстановок, якими визначається алгоритм, називають *схемою нормального алгоритму*. Підстановки схеми або нумеруються, або послідовно записуються в колонку зверху вниз чи в рядок через кому.

$$S:\begin{cases}\alpha_1 \to (\bullet)\beta_1,\\ \quad\dots\\ \alpha_n \to (\bullet)\beta_n.\end{cases}$$

Зауваження. Будь-який нормальний алгоритм може бути заданий спеціальною граф-схемою, вузли якої, крім вхідного і вихідного, містять тільки оператори підстановки. Функціонування такої граф-схеми здійснюється у відповідності з п.1)-п.4) роботи нормального алгоритму.

Слід відмітити, що застосування в схемах нормальних алгоритмів поряд із звичайними і заключних підстановок є необхідною умовою універсальності таких алгоритмів, тобто можливості реалізації ними будь-яких алгоритмів як алфавітних операторів, що задаються скінченним числом правил.

Покажемо необхідність використання заключних підстановок. Насамперед відмітимо, що коли нормальний алгоритм *U*, в схему якого не входять заключні підстановки, застосовний до заданого слова α, то виконується умова: $U(U(\alpha)) = U(\alpha)$, тобто результат дії такого алгоритму *U* на вхідне слово α є слово $U(\alpha)$, до якого уже не застосовна жодна підстановка схеми алгоритму *U.* Простим прикладом алгоритму, що не задовольняє цій умові, є алгоритм U_1, який до будь-якого слова α приписує зліва лише одну букву a: $U_1(\alpha) = a\alpha$. Тоді $U_1(U_1(\alpha)) = U_1(a\alpha) = aa\alpha$ і вказана вище умова не виконується. Отже, цей алгоритм не може бути реалізований нормальним алгоритмом, в схемі якого немає заключних підстановок. В той же час легко переконатися, що цей алгоритм реалізується нормальною схемою, яка містить одну заключну підстановку: $\Lambda \to \bullet a$. Застосування цієї підстановки до будь-якого слова α переводить його в слово $a\alpha$ і робота алгоритму на цьому припиняється.

Покажемо також, що неможливо обмежитись тільки заключними підстановками. Дійсно, дія нормального алгоритму, в схему якого входять лише заключні підстановки, складається з одноразового застосування тільки однієї з таких підстановок. А тому довжина вихідного слова $U(\alpha)$ відрізняється від довжини вхідного слова α скінченним числом n, що не залежить від довжини вхідного слова α. Це число n визначається як максимум модулів різниць між довжинами слів у лівій і правій частинах підстановок алгоритму *U.* В той же час існують алгоритми, для яких різниця між довжинами вхідного і вихідного слів залежить від довжини вхідного слова і може бути як завгодно великою. Наприклад, таким є алгоритм подвоєння слів $U_2(\alpha) = \alpha\alpha$. Отже, такий алгоритм не можна реалізувати нормальним алгоритмом із схемою, що складається тільки із заключних підстановок (див. вправу 4.18).

Отже, якщо до алгоритмічної системи, побудованої на використанні нормальних алгоритмів, пред'являти вимогу універсальності (можливості побудови нормального алгоритму, еквівалентного будь-якому наперед заданому алгоритму), то необхідною умовою такої універсальності є використання обох видів підстановок - як заключних, так і звичайних. Ця умова є також і достатньою, тобто можна сформулювати так званий принцип нормалізації (гіпотеза А.А. Маркова).

Принцип нормалізації. Для будь-якого алгоритму в довільному скінченному алфавіті A можна побудувати еквівалентний йому нормальний алгоритм над алфавітом A, або коротко: всякий алгоритм нормалізований.

По суті, принцип нормалізації уточнює поняття будь-якого алгоритму через поняття нормального алгоритму. Згідно з принципом нормалізації кожного разу, коли постає питання про можливість алгоритмізувати той чи інший процес, його можна замінити питанням про можливість реалізації цього процесу у вигляді нормального алгоритму.

Дати строге математичне доведення принципу нормалізації неможливо, оскільки поняття будь-якого алгоритму не є строго математичним. Тому до його обґрунтування слід підходити так, як це робиться при обґрунтуванні природничих законів, опираючись на практичний досвід та експеримент.

Справедливість принципу нормалізації базується на таких ідеях.

1). Всі відомі до цього часу алгоритми є нормалізованими. Оскільки на протязі довгої історії розвитку точних наук було розроблено багато різних алгоритмів, то цей факт є досить переконливим в підтвердження принципу нормалізації.

2). Всі відомі в наш час способи *композиції алгоритмів* (суперпозиція, об'єднання, розгалуження, ітерація тощо), які дають можливість будувати нові алгоритми із уже відомих, приводять знову до нормалізованих алгоритмів.

3). Виявилось, що всі відомі різні строго математичні універсальні алгоритмічні системи (нормальні алгоритми, рекурсивні функції, машини Тюрінга та ін.) рівносильні між собою, тобто для всякого алгоритму в одній алгоритмічній системі можна побудувати еквівалентний йому алгоритм в іншій алгоритмічній системі. Таким чином, всі відомі алгоритмічні системи приводять, по суті (з точністю до еквівалентності), до одного і того ж класу алгоритмів. Цей факт свідчить про те, що сучасна теорія алгоритмів охоплює надзвичайно широкий клас (якщо не всі) конструктивно визначені алфавітні оператори.

Приклад 1. Нормальний алгоритм U_1 в алфавіті $A=\{a,b\}$ заданий схемою:

$$\begin{cases} bab \to aa \\ aa \to b \\ bb \to a \end{cases}$$

Знайти результат дії цього алгоритму на слово $\alpha = abaaabb$.

Перша підстановка до α незастосовна, а друга - застосовна. Виконавши її, одержимо слово $\alpha_1 = abbabb$. До α_1 застосовною є перша підстановка. Виконавши її, дістанемо слово $\alpha_2 = abaab$. До α_2 перша підстановка незастосовна. Застосувавши до α_2 другу підстановку, матимемо слово $\alpha_3 = abbb$. Перші дві підстановки до слова α_3 будуть незастосовні. Виконавши третю підстановку, прийдемо до слова $\alpha_4 = aab$. До одержаного слова α_4 перша підстановка незастосовна. Виконавши далі послідовно другу, а потім третю підстановки схеми, одержимо результат: $U_1(abaaabb) = a$, бо жодна підстановка до слова a уже незастосовна.

Приклад 2. В алфавіті $A=\{a,b,c\}$ нормальний алгоритм U_2 заданий такою системою підстановок: $\begin{cases} ba \to ab \\ ca \to ac \\ cb \to bc \end{cases}$

Знайти $U_2(acbcab)$.

Коротко роботу алгоритму опишемо наступним чином:

$$\begin{aligned}
&\alpha = acb(ca)b, \quad ca \to ac; \quad \alpha_1 = ac(ba)cb, \quad ba \to ab; \quad \alpha_2 = a(ca)bcb, \quad ca \to ac; \\
&\alpha_3 = aa(cb)cb, \quad cb \to bc; \quad \alpha_4 = aabc(cb), \quad cb \to bc; \quad \alpha_5 = aab(cb)c, \quad cb \to bc; \\
&\alpha_6 = aabbcc.
\end{aligned}$$

До слова $\alpha_6 = aabbcc.$ жодна підстановка не застосовна, отже $U_2(acbcab) = aabbcc$.

Детальний аналіз змісту підстановок цього алгоритму приводить до висновку, що він в будь-якому слові, заданому в алфавіті $\{a,b,c\}$, впорядковує його букви в алфавітному порядку.

Приклад 3. Нехай натуральні числа записуються в алфавіті {I}. Вхідне слово має вигляд m$*$n=III$*$IIII, тобто записане в розширеному алфавіті {I, $*$}. Нормальний алгоритм U_3 заданий схемою: $\begin{cases} *I \to I* \\ * \to \wedge \end{cases}$

Знайти U_3(III$*$IIII)

Застосовуючи чотири рази першу підстановку, одержимо слово IIIIIII$*$, до якого перша підстановка уже не застосовна. Після цього спрацьовує один раз друга підстановка, яка стирає букву $*$ і робота алгоритму на цьому закінчується, тобто U_3(III$*$IIII)=IIIIIII.

Досить простий аналіз роботи цього алгоритму свідчить, що будь-яке вхідне слово виду $\alpha = m*n$ (де m і n- довільні натуральні числа) він перетворює у суму m+n, тобто наведений нормальний алгоритм реалізує суму двох натуральних чисел.

Задачі, що розглянуті в прикладах 1-3 досить прості. Задається вхідне слово α і нормальний алгоритм U , потрібно знайти $U(\alpha)$. Для цього чисто формально виконуємо оператори нормального алгоритму у відповідності з процедурою його функціонування.

Значно складнішими є задачі, в яких ставиться вимога розробити нормальний алгоритм, який розв'язує певний клас задач. Для цього потрібно на змістовному рівні досить глибоко аналізувати розв'язок поставленої задачі, пропонувати різні варіанти підстановок, що ведуть до розв'язку, і для остаточно побудованого нормального алгоритму необхідно перевірити правильність його роботи на всіх можливих варіантах вхідних даних (тобто тут виникає ще і проблема налагодження розробленого алгоритму шляхом тестування).

Приклад 4. Побудувати нормальний алгоритм U_4, який перетворює будь-яке слово α в алфавіті $A = \{a,b,c\}$ в слово abc, тобто $U_4(\alpha) = abc$.

Нехай вхідним словом буде $\alpha = bacb$. Проведемо такі змістовні міркування: на першому етапі перетворення кожну букву вхідного слова (зліва направо) зітремо, тобто замінимо на пусте слово Λ, а в кінці слово Λ замінимо на abc і процес перетворення вхідного слова на цьому завершимо. А тому шуканий алгоритм можна задати схемою:

$$\begin{cases} a \to \Lambda \\ b \to \Lambda \\ c \to \Lambda \\ \Lambda \to \bullet abc \end{cases}$$

Очевидно, що на будь-якому слові α $U_4(\alpha) = abc$.

Приклад 5. Побудувати нормальний алгоритм U_5, що перетворює будь-яке слово $\alpha = P * Q$, де *P, Q* – слова в алфавіті $A = \{a, b\}$, в слово *P*, тобто $U_5(P * Q) = P$.

Змістовно очевидно, що на початку роботи треба знищити всі букви вхідного слова, що стоять після букви $*$. Це досягається підстановками: $\begin{cases} * a \to * \\ * b \to * \end{cases}$.

Після того, як усі букви слова *Q* будуть усунені з даного вхідного слова, залишається знищити букву $*$, що реалізується підстановкою: $* \to \Lambda$. А тому шуканий алгоритм задається схемою: $\begin{cases} * a \to * \\ * b \to * \\ * \to \Lambda \end{cases}$

Приклад 6. Скласти нормальний алгоритм U_6 який перетворює будь-яке натуральне число n, записане в алфавіті *A*={I}, в частку від ділення цього числа на три.

Нехай n=IIIIIII. Замінюючи кожні три палички в цьому слові на одну палочку, ми одержимо результат. Але в перетвореному слові за часткою залишиться ще і остача, яку потрібно в кінці роботи знищити. Для відокремлення в слові одиниць частки від одиниць остачі введемо допоміжну букву $*$, яку треба записати на початку слова. Тоді, як легко переконатися, шуканий алгоритм задається схемою: $\begin{cases} * III \to I * \\ * I \to * \\ * \to \bullet \Lambda \\ \Lambda \to * \end{cases}$

На першому кроці роботи цього алгоритму застосовною до вхідного слова $n = IIIIIII$ буде тільки остання підстановка, в результаті застосування якої на початку вхідного слова буде записана буква $*$ і одержимо нове слово $n_1 = *IIIIIII$. До цього слова два рази буде застосовною перша підстановка, в результаті чого одержимо слова $n_2 = I * IIII$ і $n_3 = II * II$. До останнього слова два рази застосуємо другу підстановку, одержимо слова $n_4 = II * I$ і $n_5 = II *$. Використавши третю підстановку одержимо результат $U_6(IIIIIII) = II$.

Приклад 7. Побудувати нормальний алгоритм U_7 множення двох натуральних чисел *n* і ***m,*** записаних в алфавіті *{I}*.

Вхідним є слово $\alpha = n * m$ в розширеному алфавіті *{I, $*$}* (буква $*$ відіграє роль операції множення). Використавши дистрибутивний закон множення натуральних чисел, маємо:

$$\alpha = III \cdot II = (I + I + I) \cdot II = I \cdot II + I \cdot II + I \cdot II = IIIIII \ .$$

Виходячи з цього, потрібно ввести такі підстановки, які кожну одиницю першого числа *n* заміняють m одиницями. Ці нові одиниці потрібно якось позначити, щоб вони не змішувалися з одиницями числа *m*. Введемо підстановку $I* \to *a$, яка замість одиниці числа вводить букву а. Далі, користуючись буквою a, утворимо *m* позначених одиниць (наприклад, букв b), не знищуючи одиниць числа *m*, ввівши підстановку $aI \to Iba$. Після дії цієї підстановки одиниці слова *m* будуть змішані з буквами b. Щоб знову виділити слово *m* і позначене слово $\overline{m}$, що складається з букв b, введемо підстановку $bI \to Ib$. Введені три підстановки майже розв'язують поставлене завдання. Наприклад, слово $\alpha = III * II$ алгоритмом

$$\begin{cases} bI \to Ib \\ aI \to Iba \\ I* \to *a \end{cases}$$

перетворюється в слово $* IIbbabbabba$, в якому вже є шуканий добуток (шість букв b). Щоб завершити розробку нормального алгоритму множення двох натуральних чисел, залишається ввести підстановки, що знищують одиниці слова *m*, букви a і знак $*$, та підстановку, що замінює букви b на одиницю. Тобто введемо ще такі підстановки: $\begin{cases} * I \to * \\ a \to \Lambda \\ * \to \Lambda \\ b \to I \end{cases}$

Тепер треба ще правильно розташувати введені підстановки. Остаточно нормальний алгоритм множення двох натуральних чисел в розширеному алфавіті *{I, $*$, a, b}* задається схемою, яку запишемо впорядкованою послідовністю підстановок, розділених комою:

$$bI \to Ib, \quad aI \to Iba, \quad I* \to *a, \quad a \to \Lambda, \quad * I \to *, \quad * \to \Lambda, \quad b \to I \ .$$

<u>Запитання для самоконтролю</u>

1. Що розуміють під алгоритмом на змістовному рівні(інтуїтивне розуміння)?

2. Назвіть характеристичні властивості алгоритмів.
3. Які оператори використовуються в нормальних алгоритмах?
4. Що таке схема нормального алгоритму?
5. Дайте визначення нормального алгоритму Маркова.
6. Як визначається обробка слова нормальним алгоритмом?
7. Чим відрізняються заключні підстановки від звичайних підстановок в нормальному алгоритмі?
8. Як визначається обчислювальність функції $f:N^n \to N$ за допомогою НА?
9. Що таке НА-обчислювальна функція?
10. В чому суть принципу нормалізації?

ВПРАВИ

4.1. Натуральні числа записуються в алфавіті $A=\{I\}$, а схема нормального алгоритму містить одну підстановку: $\wedge \to \bullet I$. Яку функцію реалізує цей алгоритм?

4.2. Нормальний алгоритм заданий в алфавіті $A=\{a,b\}$ схемою: $a \to \wedge, b \to b$. Вияснити, до яких слів цей алгоритм застосовний, а до яких незастосовний.

4.3. Побудувати нормальний алгоритм, який до будь-якого слова дописує зліва дві букви a.

4.4. Задати НА, що знаходить модуль різниці двох натуральних чисел m і n записаних в алфавіті $A=\{I\}$.

4.5. Побудувати НА U для знаходження найбільшого спільного дільника двох натуральних чисел (алгоритм Евкліда), заданих в алфавіті $A=\{I\}$.

4.6. Нехай задано два алфавіти $A=\{a,b\}$ і $B=\{\alpha,\beta\}$, між буквами яких встановлена бієктивна відповідність: $a \leftrightarrow \alpha, b \leftrightarrow \beta$. Скласти НА (так званий алгоритм перекладу), який будь-яке слово *P* в алфавіті *A* переводить в слово $\overline{P}P$, де $\overline{P}$ - слово у відповідному алфавіті *B*.

4.7. В алфавіті $A=\{a,b\}$ задане слово α. Побудувати НА в розширеному алфавіті $A \cup \{I\}$, який би знаходив довжину слова α.

4.8. Знайти НА U, який для будь-якого натурального числа *n,* записаного в двійковій системі числення, тобто в алфавіті $A=\{0,1\}$ знаходить наступне число в цій же системі числення, тобто $U(n)=n+1$.

4.9. Побудувати НА U, що перетворює будь-яке слово $P*Q$, де P,Q -слова в алфавіті $A=\{a,b,c\}$, в слово Q, тобто $U(P*Q)=Q$.

4.10.Скласти НА, який слово $P*Q*R$ перетворює в слово P, де P, Q, R - слова в алфавіті $A=\{a,b\}$.

4.11.Знайти НА U, який перетворює будь-яке натуральне число, записане в алфавіті $A=\{I\}$, в остачу від ділення цього числа на 3.

4.12.Побудувати НА, що будь-яке натуральне число, записане в алфавіті $A=\{I\}$ перетворює в те ж саме натуральне число, записане в двійковій системі числення в алфавіті $B=\{0, 1\}$.

4.13.Побудувати НА, що перетворює всяке натуральне число n, записане в алфавіті $A=\{I\}$, в натуральне число kn, де k - фіксоване натуральне число в тому ж алфавіті A.

4.14.Побудувати НА U, що реалізує функцію натурального аргументу $f(n)=2n+1$.

4.15.Побудувати НА, який всяке слово в алфавіті $A=\{I\}$ перетворює в частку від ділення цього числа на два, записану в тому ж алфавіті A.

4.16.Скласти НА, що перетворює кожне натуральне число n в алфавіті $A=\{I\}$ в цілу частину числа $\frac{2n}{3}$, тобто перетворює число n в $\left[\frac{2n}{3}\right]$.

4.17.Побудувати НА, який слово $m*n$ (m і n слова в алфавіті $A=\{I\}$) перетворює в слово $\left[\frac{m+n}{3}\right]$, записане в тому ж алфавіті A.

4.18.Знайти НА U подвоєння слів, тобто алгоритм, який всяке слово P в алфавіті $A=\{a,b\}$ перетворює в слово PP, тобто $U(P)=PP$.

4.3. РЕКУРСИВНІ ФУНКЦІЇ

Історично першою алгоритмічною системою є система, запропонована в 1936 році американським математиком А. Черчем, яка базується на використанні конструктивно означених арифметичних (цілозначних) функцій, які одержали назву *рекурсивних функцій.* Застосування подібних функцій в теорії алгоритмів ґрунтується на ідеї нумерації слів в довільному скінченному алфавіті послідовними натуральними числами. Найбільш просто таку нумерацію можна здійснити, розташовуючи слова в будь-якому алфавіті у порядку зростання їх довжини, а слова, які мають однакову довжину, впорядкувати довільно, наприклад, в лексикографічному (алфавітному) порядку. Тоді кожному слову буде відповідати його номер, тобто між сукупністю слів та множиною натуральних чисел встановлюється бієктивна відповідність. При цьому кожному з алфавітних операторів відповідатиме певна функція $y = f(x)$, аргумент якої, як і сама функція, приймає цілі додатні значення, а сама функція називається *арифметичною*. Арифметична функція називається *загально арифметичною*, якщо її область визначення співпадає з усією множиною натуральних чисел, в протилежному випадку функцію називають *частково арифметичною*. Наприклад, функція $z = x(y+7)$ є загально арифметичною, а функція $y = \sqrt{x+2}$ - частково арифметична.

Функція, для якої існує алгоритм обчислення її значень, називають *обчислювальною функцією.* Оскільки загальне поняття алгоритму не є математично точним, то таким же є і поняття обчислювальної функції, яке потрібно уточнити.

Серед обчислювальних функцій виділимо найбільш прості функції, які називають *елементарними арифметичними функціями* або *базовими*.

1). $O^n(x_1, x_2, ..., x_n) = 0$ - константа нуль або нуль-функція.

2). $I_k^n(x_1, x_2, ..., x_n) = x_k$ $(k \le n)$ - функція-проектор або функція вибору *k*-го аргументу.

3). $S(x) = x + 1$ - функція слідування.

Аргументи у всіх цих функцій - натуральні числа. Число 0 також вважаємо натуральним числом.

Із базових арифметичних функцій за допомогою трьох операцій - *суперпозиції (підстановки)*, *примітивної рекурсії та операції найменшого кореня (операція мінімізації)* будуються більш складні обчислювальні функції. При цьому в обчислювальних функціях можна використовувати додаткові, так звані фіктивні аргументи, від яких функція фактично не залежить.

Наприклад, функцію слідування можна розглядати як функцію двох (і більше) аргументів: $\sigma(x, y) = x + 1 = S(x)$.

1). Операція *суперпозиції (підстановки)* дає можливість із відомих функцій $\varphi_1(x_1, ..., x_n)$, $\varphi_2(x_1, ..., x_n), \ldots,$ $\varphi_k(x_1, ..., x_n)$, $f(x_1, ..., x_k)$ побудувати нову функцію: $F(x_1, x_2, ..., x_n) = f(\varphi_1(x_1, ..., x_n), \varphi_2(x_1, ..., x_n), ..., \varphi_k(x_1, ..., x_n))$.

Приклад 1. З нуль-функції $O(x)=0$ і функції слідування $S(x)=x+1$ підстановкою одержується функція $S(O(x))=1$ - константа одиниця, або *константна функція* $c_1 = S(O(x)) = 1$. Аналогічно можна одержати всі інші константи (натуральні числа):

$$k = c_k^n(x_1, x_2, ..., x_n) = \underbrace{S(S...S}_{k \ раз}(O^n(x_1, x_2, ..., x_n))...).$$

Приклад 2. Окремими випадками операції суперпозиції є *операції перестановки аргументів та ідентифікації аргументів*.

Так, функція $f(x,y)=\varphi(y,x)$ утворюється з функції $\varphi(x,y)$ підстановкою: $f(x,y)=\varphi(I_2^2(x,y), I_1^2(x,y))$, де $I_2^2(x,y)=y, I_1^2(x,y)=x$ - функції-проектори на другу та першу змінні *y* та *x* відповідно.

Функція $f(x)=\varphi(x,x)$ утворюється з функції $\varphi(x,y)$ підстановками: $f(x)=\varphi(I_1^2(x,y), I_1^2(x,y))$.

2). Операція *примітивної рекурсії* дає можливість будувати *n*-місну арифметичну функцію (функцію від *n* аргументів) за двома заданими функціями, одна з яких (*n-1*)-місна, а друга – (*n+1*)-місна. Ця операція визначається такими двома співвідношеннями:

$$\begin{cases} f(x_1, x_2, ..., x_{n-1}, 0) = g(x_1, x_2, ..., x_{n-1}), \\ f(x_1, x_2, ..., x_{n-1}, x_n + 1) = h(x_1, x_2, ..., x_{n-1}, x_n, f(x_1, x_2, ..., x_n)). \end{cases}$$

де функції $g(x_1, x_2, ..., x_{n-1})$ і $h(x_1, x_2, ..., x_{n+1})$ задані, а функція $f(x_1, x_2, ..., x_{n-1}, x_n)$ визначається цими співвідношеннями, які називають *схемою примітивної рекурсії*. Кажуть, що $f(x_1, x_2, ..., x_{n-1}, x_n)$ одержана за схемою примітивної рекурсії з функцій $g(x_1, x_2, ..., x_{n-1})$ і $h(x_1, x_2, ..., x_{n+1})$.

Приклад 3. Нехай задана константа (константна функція $k=g(x)$)і функція двох аргументів $h(x,y)$. Схема примітивної рекурсії:

$$\begin{cases} f(0) = k, \\ f(x+1) = h(x, f(x)). \end{cases}$$

визначає функцію одного аргументу $f(x)$, значення якої можна послідовно обчислити, якщо відомі значення $h(x,y)$, дійсно:

$$f(1) = h(0, f(0)) = h(0,k),$$
$$f(2) = h(1, f(1)) = h(1, h(0,k)),$$
$$f(3) = h(2, f(2)) = h(2, h(1, h(0,k))),$$

….

Очевидно, що за допомогою операції примітивної рекурсії при заданих функціях g і h функція f визначається однозначно.

Приклад 4. Покажемо, що функція додавання двох натуральних чисел визначається схемою примітивної рекурсії за допомогою функцій $I_1^1(x) = x$ та функції слідування $S(x, y, z) = z + 1$, в якої аргументи x і y є фіктивними. Дійсно, маємо:

$$\begin{cases} f(x,0) = x + 0 = I_1^1(x), \\ f(x, y+1) = x + (y+1) = (x+y) + 1 = S(x, y, f(x, y)). \end{cases}$$

Функції, які будуються з базових арифметичних функцій за допомогою операцій суперпозиції і примітивної рекурсії, застосованих скінченне число разів у довільній послідовності, називаються *примітивно-рекурсивними функціями.*

Оскільки базові функції всюди визначені, а операції підстановки і примітивної рекурсії зберігають всюди визначеність побудованих функцій, то всі примітивно-рекурсивні функції всюди визначені.

Більшість відомих арифметичних функцій належать до класу примітивно-рекурсивних функцій. Ми вже мали нагоду переконатися (див. приклад 4) що функція $f(x, y) = x + y$ є примітивно рекурсивною, оскільки вона одержується за схемою примітивної рекурсії із базових арифметичних функцій $I_1^1(x)$ і $S(x, y, z) = z + 1$, які є примітивно рекурсивними.

Приклад 5. Довести, що наступні функції є примітивно рекурсивними:

а) $f(x, y) = x \cdot y$.

Ця функція задається такою схемою примітивної рекурсії:

$$\begin{cases} f(x,0) = x \cdot 0 = O(x), \\ f(x, y+1) = x(y+1) = x + x \cdot y = h(x, y, f(x, y)). \end{cases}$$

де функції $O(x)$ і $h(x, y, z) = x + z$ (сума з фіктивним аргументом y) - примітивно рекурсивні, а значить і $f(x, y) = x \cdot y$ є примітивно рекурсивною функцією.

б) $f(x) = \begin{cases} x - 1, \textit{ якщо } x \geq 1; \\ 0, \textit{ якщо } x = 0. \end{cases}$

Цю функцію позначають символом $x \dot{-} 1$. Для неї маємо:

$$\begin{cases} 0 \dot{-} 1 = 0 = O(x), \\ (x+1) \dot{-} 1 = x = I_1^2(x, y). \end{cases}$$

Таким чином, функція $f(x) = x \dot- 1$ визначається примітивною рекурсією за допомогою примітивно рекурсивних функцій $O(x)$ та $h(x,y) = I_1^2(x,y)$, а тому функція $f(x) = x \dot- 1$ є примітивно рекурсивною.

в) $f(x,y) = x \dot- y = \begin{cases} x - y, x > y; \\ 0, x \le y. \end{cases}$

Ця функція задається такою схемою примітивної рекурсії:

$$\begin{cases} x \dot- 0 = x = I_1^1(x), \\ x \dot- (y+1) = (x \dot- y) \dot- 1 = h(x, y, f(x,y)), \end{cases}$$

де одномісна функція $I_1^1(x)$ та трьохмісна функція $h(x,y,z) = z \dot- 1$ (в якої аргументи x і y є фіктивними) є примітивно рекурсивними.

г) $sg(x) = \begin{cases} 1, x > 0; \\ 0, x = 0. \end{cases}$

Досить переконатися, що:

$$\begin{cases} sg(0) = 0 = O(0), \\ sg(x+1) = 1 = h(x,y), \end{cases}$$

де $h(x,y) = c_1^2(x,y)$ – константна функція 1 від 2-ох змінних.

Отже, $sg(x)$ одержується схемою примітивної рекурсії з констант 0 та I, які є примітивно рекурсивними функціями.

д) $\sigma(x) = x!$.

Вважаючи, що $\sigma(0) = 0! = 1$, маємо:

$$\begin{cases} \sigma(0) = 1, \\ \sigma(x+1) = x! \cdot (x+1) = h(x, \sigma(x)), \end{cases}$$

де $h(x,y) = f(s(x), \sigma(x))$ - добуток, звідки слідує, що функція $\sigma(x) = x!$ є примітивно рекурсивною.

Приклад 6. Нехай $\max(x,y)$ - двомісна функція, значення якої дорівнює більшому з пари чисел $\{x,y\}$. Довести, що $\max(x,y)$ є примітивно рекурсивною функцією.

Доведення, що функція $\max(x,y)$ є примітивно рекурсивною слідує з того, що вона задається наступною формулою:

$$\max(x,y) = x \cdot sg(x \dot- y) + y \cdot sg(x - y).$$

Приклад 7. Розглянемо функцію $ent\left(\frac{x}{y}\right)=\left[\frac{x}{y}\right]$ - частка від ділення числа x на число y, причому домовимось, що $\left[\frac{x}{0}\right]=x$. Довести, що функція $\left[\frac{x}{y}\right]$ – примітивно рекурсивна.

За означенням $\left[\frac{x}{y}\right]$ дорівнює найбільшому числу n такому, що $ny \le x$, а $x<(n+1)y$. Всі різниці $iy \dot{-} x$ дорівнюють нулю при $i \le n$ і відмінні від нуля тоді, коли $n+1 \le i \le x$. Таким чином, якщо i змінюється від 1 до x, то вираз $iy \dot{-} x$ дорівнює нулю точно n разів. Отже, $\left[\frac{x}{y}\right]=\sum_{i=1}^{x}\overline{sg}(iy \dot{-} x)$. Враховуючи рівність $\left[\frac{x}{0}\right]=0$ і одержану формулу, легко вказати схему примітивної рекурсії, якою визначається функція $\left[\frac{x}{y}\right]$.

Приклад 8. Позначимо остачу від ділення числа x на число y через $rest(x,y)$, причому за її означенням $rest(x,0)=x$. Чи буде функція $rest(x,y)$ примітивно рекурсивною?

Позитивна відповідь на поставлене питання слідує з того, що функція $rest(x,y)$ задається через відомі примітивно рекурсивні функції формулою:

$$rest(x,y)=x \dot{-}\left(y\cdot\left[\frac{x}{y}\right]\right).$$

Приклад 9. Визначимо двомісний предикат $div(x,y)$ у такий спосіб:

$$div(x,y)=\begin{cases}1, & x \vdots y,\\ 0, & x \,\overline{\vdots}\, y.\end{cases}$$

Чи можна вважати цей предикат примітивно рекурсивною функцією?

Так, тому що $div(x,y)=\overline{sg}(rest(x,y))$.

Приклад 10. Нехай $d(x)$ - число всіх різних дільників числа x. Довести, що функція $d(x)$ є примітивно рекурсивною.

Виходячи з означення функції $div(x,y)$, можемо записати, що $div(x,x)=1$ і $div(x,y)=0$, якщо $x<y$. Кожного разу, коли x ділиться на i, $div(x,i)=1$. Тому $d(x)=\sum_{i=0}^{x}div(x,i)$.

Остання формула дає можливість легко записати схему примітивної рекурсії, якою і підтверджується примітивна рекурсивність функції $d(x)$

Приклад 11. Характеристичну функцію $\lambda(x)$ множини *М* визначимо таким способом:

$$\lambda(x) = \begin{cases} 1, \, x \in M, \\ 0, \, x \notin M. \end{cases}$$

Довести, що характеристична функція множини всіх простих чисел є примітивно рекурсивною.

Для цього досить показати, що ця функція є суперпозицією відомих примітивно рекурсивних функцій, а саме:

$$\lambda(x) = \overline{sg}\left(\left|d(x) - 2\right|\right).$$

Як показують приклади 1-11 багато відомих арифметичних функцій належить до класу примітивно рекурсивних функцій. Але цей клас функцій не охоплює всіх обчислювальних арифметичних функцій, що можна довести такими теоретико-множинними міркуваннями.

Як легко бачити, множина примітивно рекурсивних функцій зліченна. Множина ж обчислювальних теоретико-числових функцій виду $[tn]$, де *n* - фіксоване натуральне число, а *t* приймає всі дійсні невід'ємні значення, утворює множину потужності континуум. Отже, існують. такі обчислювальні функції, які не є примітивно рекурсивними.

Досить просто навести приклад такої обчислювальної функції, яка не є примітивно рекурсивною. Справа в тому, що всі примітивно рекурсивні функції всюди визначені, а, наприклад, функція однієї натуральної змінної $f(x)$, значенням якої є найменше *z*, яке задовольняє рівнянню x+1+z=0, ніде невизначена.

Для розширення класу примітивно рекурсивних функцій введемо нову операцію, так звану операцію *найменшого кореня або операцію мінімізації*.

Операція *найменшого кореня (мінімізації)* дає можливість визначити функцію $f(x_1, x_2, ..., x_n)$ від *n* змінних за допомогою відомої арифметичної функції $g(x_1, x_2, ..., x_n, y)$ від (*n*+1)-ї змінної. Для будь-якого набору значень змінних $x_1 = a_1, ..., x_n = a_n$ в ролі відповідного значення функції $f(a_1, a_2, ..., a_n)$ приймається найменше натуральне число *y*, для якого $g(a_1, a_2, ..., a_n, y) = 0$. Позначається ця операція так: $\mu_y[g(a_1, a_2, ..., a_n, y) = 0]$, де μ_y символ операції найменшого кореня, а в квадратних дужках стоїть предикат, який набуває значення 1 при $y = \beta$. Очевидно, що значення β є функцією $f(a_1, a_2, ..., a_n)$. Отже, шукана функція $f(x_1, x_2, ..., x_n)$ визначається так:

$$f(x_1, x_2, ..., x_n) = \mu_y \left[g(x_1, x_2, ..., x_n, y) = 0 \right].$$

При цьому, якщо не існує таких значень y, при яких $g(x_1, x_2, ..., x_n, y) = 0$, то функція $f(x_1, x_2, ..., x_n)$ вважається невизначеною на відповідному наборі значень $x_1, x_2, ..., x_n$. Крім того вважається, що функція $f(x_1, x_2, ..., x_n)$ невизначена на такому наборі значень аргументів $x_1 = a_1, ..., x_n = a_n$, для якого існує корінь рівняння $g(a_1, a_2, ..., a_n, y) = 0$, проте хоча б для одного значення $\gamma\ (0 \le \gamma < \beta)$ функція $g(a_1, a_2, ..., a_n, \gamma)$ невизначена.

Функції, які будуються з базових арифметичних функцій за допомогою скінченного числа операцій суперпозиції, примітивної рекурсії та мінімізації, називаються *частково рекурсивними функціями* (ЧРФ).

Всюди визначені частково рекурсивні функції називаються *загально рекурсивними функціями* (ЗРФ).

Очевидно, що кожна ЧРФ є обчислювальною. Обернена твердження складає зміст *гіпотези Черча*: будь-яка частково обчислювальна функція є ЧРФ.

Оскільки інтуїтивне поняття алгоритму було ототожнене з поняттям обчислювальної функції, то гіпотеза Черча може бути сформульована і в такому вигляді: будь який алгоритм може бути реалізований деякою ЧРФ.

Обґрунтування гіпотези Черча проводиться аналогічно обґрунтуванню гіпотези Маркова.

Після здійснення нумерації вхідних і вихідних слів будь-який нормальний алгоритм може бути реалізований деякою ЧРФ. Навпаки, всякий алгоритм, що реалізується за допомогою ЧРФ, виявляється еквівалентним деякому нормальному алгоритму. Отже, має місце таке твердження: алгоритм нормалізований тоді і тільки тоді, коли його можна реалізувати у вигляді ЧРФ.

Таким чином, на основі уточнення поняття обчислювальних функцій як частково рекурсивних функцій будується загальна теорія алгоритмів, яка рівносильна розглянутій раніше теорії нормальних алгоритмів.

Приклад 12. Показати, що застосування операції найменшого кореня до функції $g(x_1, x_2, ..., x_n, y)$, коли вона є функцією одного аргументу, тобто $g(y)$ породжує константу (константну функцію), тобто $\beta = \mu_y[g(y) = 0]$.

Дійсно, щоб знайти y, таке що $\mu_y[g(y) = 0]$, потрібно знайти корені рівняння $g(y) = 0$ і в ролі β взяти мінімальний, тобто β буде константа. При цьому, якщо не існує кореня рівняння $g(y) = 0$, то β буде всюди невизначеною функцією.

Приклад 13. За допомогою оператора мінімізації знайти функцію $f(x, y)$, якщо $g(x, y, z) = x + y - z$.

Найменше значення *z,* при якому для заданих *x*, *y* функція *g* дорівнює нулю, є сума *x*+*y*. Для всіх значень *Z*, менших *x*+*y*, функція *g* визначена і не дорівнює нулю. Отже $f(x, y) = x + y = \mu_z(x + y - z = 0)$.

Приклад 14. За допомогою оператора мінімізації побудувати арифметичну функцію двох змінних $f(x, y)$, якщо задана часткова арифметична функція $g(x, y, z) = x - y + \sqrt{z}$.

Покладемо, наприклад, *x*=5, *y*=9. Знайдемо z таке, що $\mu_z(5 - 9 + \sqrt{z} = 0)$. Вибираючи послідовно *z*=0,1,2,..., перевіркою встановлюємо, що $\mu_z(5 - 9 + \sqrt{z} = 0) = 16$, при цьому $z = (y - x)^2$. Отже, шуканою функцією буде $f(x, y) = \mu_z(x - y + \sqrt{z} = 0) = (y - x)^2$.

Приклад 15. Для функції $f(x, y) = x - y$ знайти таку функцію $g(x, y, z)$, що $f(x, y) = \mu_z[g(x, y, z) = 0]$.

Такою функцією, очевидно, буде $g(x, y, z) = x - y - z$, адже $f(x, y) = x - y = \mu_z(x - y - z = 0)$.

Запитання для самоконтролю

1. Яка функція називається арифметичною; всюди визначеною; частково визначено?
2. Яка функція називається обчислювальною?
3. Які функції належать до базових арифметичних функцій?
4. Що називають операцією суперпозиції?
5. Що називають операцією примітивної рекурсії та що являє собою схема примітивної рекурсії?
6. Що називають операцією мінімізації?

7. Яка функція називається примітивно рекурсивною, загально рекурсивною, частково рекурсивною?
8. Як формулюється гіпотеза Черча?

ВПРАВИ

4.19.Чи буде примітивно рекурсивною функція $|x-y|$?

4.20.Довести, що кожна функція, яка тотожно дорівнює натуральному числу *n,* є примітивно рекурсивною.

4.21.Нехай функція $\overline{sg}(x)$ визначена за допомогою таких рівностей:

$$\overline{sg}(x)=\begin{cases}0, & x>0,\\ 1, & x=0.\end{cases}$$

Чи буде $\overline{sg}(x)$ примітивно рекурсивною функцією?

4.22.Довести таку рівність: $\overline{sg}(x)=1\dot{-}x$.

4.23.Довести, що функція $f(x)=a^x$ є примітивно рекурсивною.

4.24.Нехай $\min(x,y)$ - функція двох аргументів, що дорівнює меншому з пари $\{x,y\}$, або одному з них, якщо *x*=*y*. Довести, що функція $\min(x,y)$ є примітивно рекурсивною.

4.25.Функція $\pi(x)$ визначає число всіх простих чисел, які не перевищують натурального числа *x*. Чи є $\pi(x)$ примітивно рекурсивною функцією?

4.26.Показати, що якщо функція $g(x_1,x_2,...,x_n)$ є примітивно рекурсивною, то наступні функції також примітивно рекурсивні:

а) $f_1(x_1,x_2,...,x_n)=\sum_{i=0}^{x_n} g(x_1,x_2,...,x_{n-1},i)$;

б) $f_2(x_1,x_2,...,x_n)=\prod_{i=0}^{x_n} g(x_1,x_2,...,x_{n-1},i)$.

4.27.Довести, що характеристична функція $\lambda(x)$ довільної скінченної множини натуральних чисел є примітивно рекурсивною функцією.

4.4. МАШИНА ТЬЮРИНГА

Розглянемо ще одне уточнення змістовного поняття алгоритму, так звану машину Тьюрінга (МТ), яке було запропоноване в 1937 році. МТ – це така алгоритмічна система, в якій правила, що визначають дію алгоритма, побудовані за командно-адресним принципом, аналогічно сучасним комп'ютерам.

МТ має два скінченних алфавіти: *зовнішній* $A=\{a_0,a_1,...,a_n\}$ і *внутрішній* $Q=\{g_0,g_1,...,g_n\}$. У зовнішньому алфавіті записується вхідна, проміжна та вихідна інформація. Внутрішній алфавіт призначений для позначення (кодування) так званих *внутрішніх станів* машини.

Вхідна інформація записується на нескінченній в обидві сторони стрічці пам’яті, яка розбита на окремі комірки, в кожній з яких записується одна буква. Вважається, що серед букв зовнішнього алфавіту є і пуста буква, яку будемо позначати символом Λ. Пусту букву, як правило, писати не будемо.

МТ має зчитуючий елемент (ЧЕ), який будемо зображати стрілкою $\uparrow$. Читаючий елемент в кожний такт роботи МТ фіксує одну комірку інформаційної стрічки.

Серед букв внутрішнього алфавіту (станів) $Q=\{g_0, g_1, ..., g_n\}$ виділяється початковий стан g_0 і, так званий, заключний стан, який означає зупинку машини. Його будемо позначати символом !.

Елементарними операціями, які може виконувати МТ, є такі:

1) зчитування букви a_i, записаної в комірці, що розглядається ЧЕ;

2) заміна букви a_i на $a_r \in A$ (можливий варіант, що *i*=*r*);

3) зсув ЧЕ вздовж стрічки на одну комірку вправо (П) або вліво (Л) або зсуву взагалі немає (Н). в останньому випадку ЧЕ продовжує оглядати ту ж саму комірку;

4) перехід зі стану g_j, в якому машина перебувала на початку роботи команди, в стан g_k (можливий варіант, що *j*=*k*).

Називатимемо *конфігурацією МТ* зображення стрічки пам'яті з розташованими на ній буквами зовнішнього алфавіту і відміченою ЧЕ (стрілкою) коміркою та фіксацією стану, в якому знаходиться машина. Наприклад, конфігурація може мати такий вигляд:

Λ	a_2	a_1	a_3	a_1	Λ
		$\uparrow g_j$			

або $\Lambda a_2 g_j a_1 a_3 a_1 \Lambda$.

Реалізація алгоритму в машині Тьюрінга відбувається за допомогою програми роботи цієї машини, яка являє собою набір команд – п'ятірок символів $a_i g_j a_r g_k P$. Ця п'ятірка символів означає, що ЧЕ, знаходячись в стані g_j і сприймаючи записану в клітці стрічки букву a_i замінює її новою буквою a_r (можливий варіант, що *i*=*r*), машина переходить в новий стан g_k (можливий варіант, що *j*=*k*), а ЧЕ зсувається вздовж стрічки на одну клітку в залежності від значення Р, де Р={Л, П, Н} - одна з команд зсуву ЧЕ.

На кожному такті роботи МТ інформація обробляється в так званому логічному блоці, який має два входи - a_i і g_j, і три виходи: a_r, g_k і Р. Очевидно, що логічний блок реалізує функцію $f(a_i, g_j) = a_r g_k P$, область визначення і область значень якої скінченні, а тому цю функцію зручно задавати у вигляді прямокутної таблиці, в рядках якої записані букви зовнішнього алфавіту, а в стовпчиках - букви внутрішнього алфавіту (стани). В кожній клітці такої таблиці будемо записувати відповідну вихідну трійку символів. Цю таблицю називають функціональною схемою машини Тьюрінга (ФС). Її заданням робота МТ визначається однозначно.

Приклад 1. Побудувати МТ (скласти її функціональну схему), яка реалізує функцію слідування *S(n)=n+1,* де *n* - натуральне число, записане в алфавіті *{I}*.

Зовнішнім алфавітом цієї машини буде алфавіт *A={I}*. Нехай початкова конфігурація машини має вигляд: $\Lambda g_0 I^n \Lambda$, тобто в початковому стані g_0 ЧЕ машини оглядає комірку, в якій записана перша зліва одиниця вхідного числа.

Змістовно можна запропонувати один з таких алгоритмів: одиниця залишається без зміни, ЧЕ зсувається вправо до тих пір, поки не зустріне першу справа пусту клітку, в яку потрійно записати одиницю. Результат отримано, машину треба зупинити.

Відповідна функціональна схема має вигляд:

A \ *Q*	g_0
I	*I*П g_0
Λ	*I*Н!

Будемо вважати, що МТ має стандартний початок роботи, якщо перед початком роботи ЧЕ машини оглядає першу зліва пусту клітку, що передує вхідному слову, і стандартний кінець роботи - після закінчення роботи машина повертається в стандартний початок. Тоді функціональна схема, що реалізує функцію слідування, запишеться так:

A \ Q	g_0	g_1	g_2
I		*I*П g_1	*I*Л g_2
Λ	П g_1	*I*Л g_2	ΛН!

Домовимось про деякі спрощення в запису функціональних схем: можна відмовитись від повного запису вихідної трійки $a_r g_k P$, опускаючи знаки a_i і g_j, які не відрізняються від відповідних вхідних знаків, а також не будемо в схемі записувати символ Н, який вказує на відсутність зсуву. З урахуванням цих спрощень попередня схема набуде такого вигляду:

A \ Q	g_0	g_1	g_2
I		П	Л
Λ	Пg_1	IЛg_2	!

Приклад 2. Побудувати МТ, що реалізує додавання двох натуральних чисел *m* і *n*, записаних в алфавіті *{I}*. Наприклад, вхідним є слово $\alpha = III * II$, а початкова конфігурація нехай буде такою: $g_0 \Lambda I^3 * I^2 \Lambda$.

Змістовно, суму двох натуральних чисел $n + m = III * II = IIIII$ можна, зокрема, одержати таким способом. В числі *n* зліва направо стираємо послідовно одиниці, записуючи їх справа від *m*, в результаті чого одержимо слово $* IIIII$. Далі в цьому слові треба зітерти букву $*$ і зупинитись.

Алгоритм додавання двох натуральних чисел при вказаній вище початковій конфігурації реалізується наступною ФС:

A \ Q	g_0	g_1	g_2
I	ΛПg_1	IПg_1	IЛg_2
Λ	ΛПg_0	IНg_2	ΛНg_0
$*$	ΛН!	$*$Пg_1	$*$Лg_2

Від цієї функціональної схеми досить просто перейти до відповідної функціональної схеми із стандартним початком та стандартним кінцем роботи, для цього потрібно ввести додаткові стани і поповнити ФС новими командами.

Приклад 3. Нехай маємо два алфавіти $A = \{a, b\}$ і $\Gamma = \{\alpha, \beta\}$, між буквами яких, встановлена бієктивна відповідність. Побудувати машину Тьюрінга, яка переводить слово в алфавіті *A* в слово в алфавіті *Г* (алгоритм перекладу).

ФС цього алгоритму із стандартними початком і кінцем роботи має вигляд:

A \ Q	g_0	g_1	g_2
a		αП	
b		βП	
α			Л

β			Л
Λ	П g_1	Л g_2	!

В пустих клітках функціональної схеми може буди записана будь-яка інформація, яка на роботу МТ не впливає.

Приклад 4. Побудувати МТ, що подвоює натуральні числа, записані в алфавіті $\{I\}$.

Нехай початкова конфігурація така: $\Lambda I^{n-1} g_0 I \Lambda$. Змістовно кожну паличку будемо замінювати по черзі на зірочку, дописуючи в кінці слова ще одну зірочку. В результаті ми одержимо $2n$ зірочок. На останньому етапі кожну зірочку замінимо на паличку і зупинимось. ФС шуканої МГ матиме такий вигляд:

A \ Q	g_0	g_1	g_2
I	$*$ П g_1		IН g_0
$*$	IП g_0	$*$ П g_1	$*$ Л g_2
Λ		$*$ Л g_2	!Н

Приклад 5. Побудувати МТ, яка будь-яке натуральне число $n > 2$ перетворює в число n-2.

Нехай початкова конфігурація така: $g_0 \Lambda I^n \Lambda$, тобто машина знаходиться в стандартному початку роботи. Складемо ФС, що реалізує вказаний алгоритм і має стандартний кінець роботи:

A \ Q	g_0	g_1	g_2	g_3
I		Λ П g_2	Λ П g_3	Л g_3
Λ	П g_1			!

Приклад 6. Нехай в алфавіті $A = \{a, b, *\}$ задане вхідне слово $\alpha = P * Q$, де P, Q- слова в алфавіті $\{a, b\}$. Побудувати таку МТ, що МТ$(P * Q) = Q$.

В ролі початкової конфігурації виберемо таку: $\Lambda g_0 bab * aab \Lambda$.

ФС шуканої МТ така:

A \ Q	g_0
a	Λ П g_0

b	$\Lambda \Pi g_0$
$*$	$\Lambda H g_0$
Λ	!

Наведені приклади алгоритмів свідчать про те, що багато відомих алгоритмів задаються машиною Тьюрінга як точним математичним об'єктом. Виникає питання, а чи будь-який алгоритм (у змістовному розумінні) може бути заданий деякою МТ. Відповідь на це дає *гіпотеза Тьюрінга*: для будь-якого алгоритму існує МТ, що його реалізує.

Очевидно, що гіпотеза Тьюрінга не є математичкою теоремою. Її обґрунтування аналогічне обґрунтуванню принципу нормалізації Маркова та гіпотези Черча. До того ж, всі розглянуті нами уточнення поняття алгоритму (нормальні алгоритми, частково рекурсивні функції та машини Тьюрінга) еквівалентні між собою.

Важливе теоретичне значення має факт існування в кожній алгоритмічній системі, в якій поняття алгоритму є математично точним, так званого *універсального алгоритму*, який може виконувати роботу будь-якого конкретного алгоритму.

Зокрема, можна побудувати універсальну машину Тьюрінга (УМТ), яка може виконувати роботу будь-якої конкретної МТ. Для цього на вхід УМТ потрібно подати два слова, одне з яких є кодом ФС конкретної МТ, а друге - код її вхідного слова в деякому стандартному, наприклад, двійковому алфавіті.

Із існування УМТ слідує теоретичний висновок (30-і роки ХХст.) про можливість побудови програмного автомата, який би виконував роботу будь-якого алгоритму. Такими автоматами є сучасні комп'ютери.

На закінчення відмітимо, що тепер, при наявності точного поняття алгоритму питання про неіснування алгоритму розв'язку тієї чи іншої задачі (масової проблеми) стає математичною теоремою. Доведення теореми про алгоритмічну нерозв'язність певного класу задач означає, що не існує і ніким, і ніякими засобами не можна побудувати єдиного алгоритму, яким розв'язуються всі задачі даного класу.

На сьогодні уже відомо багато алгоритмічно нерозв'язних проблем, зокрема, такими є:

1) проблема розпізнавання самозастосовності алгоритмів;

2) проблема розпізнавання анулювання для будь-якого алгоритму;

3) проблема розпізнавання застосовності алгоритму до того чи іншого слова;

4) комбінаторна проблема Е.Поста;

5) проблема тотожності слів для півгруп (із скінченним числом твірних елементів і скінченним числом визначальних співвідношень);

6) проблема тотожності слів для груп;

7) проблема представлення для матриць;

8) 10-а проблема Гільберта та ін.

Алгоритмічна нерозв'язність всіх вказаних проблем доводиться, виходячи із припущення про справедливість гіпотез Черча, Тьюрінга та Маркова.

Існування алгоритмічно нерозв'язних проблем означає, що при пошуку алгоритму розв'язку тієї чи іншої проблеми треба мати на увазі. що такого алгоритму можливо взагалі не існує. Тому поряд із спробами побудови такого алгоритму треба одночасно прагнути довести його неіснування.

Алгоритмічна нерозв'язність задач того чи іншого класу зовсім не означає неможливості розв'язати будь-яку конкретну задачу із цього класу. Мова йде про неможливість розв'язування всіх задач даного класу одним і тим же методом. Цілком можливо, що існують алгоритми для розв'язування окремих підкласів таких задач. Тому в таких випадках проблема формулюється стосовно більш вузького класу задач. Наприклад, якщо проблема тотожності слів, як відомо, є алгоритмічно нерозв'язною для класу всіх скінченно породжених груп, то ця проблема тотожності може ставитись для окремих підкласів груп, зокрема, скінченно породжених абелевих, нільпотентних, циклічних, локально-скінченних і т.д. груп. Головне тут полягає в тому, щоб відшукати найбільш широкий клас груп, для якого проблема тотожності розв'язується позитивно.

Пошук максимально широких класів задач, для яких та чи інша проблема розв'язується позитивно, є однією із важливих задач розвитку сучасної математики.

Запитання для самоконтролю

1. Дайте визначення машини Тьюрінга.
2. Назвіть основні складові елементи МТ?
3. Як визначається команда МТ.
4. Що таке функціональна схема МТ?

5. Що таке конфігурація МТ?
6. Що таке стандартна початкова конфігурація МТ? Фінальна конфігурація МТ?
7. Як змінюється конфігурація МТ при виконанні команди МТ відповідного типу?
8. Дайте визначення еквівалентних МТ.
9. Що такє МТ-обчислювана функція?
10. Сформулюйте гіпотезу Тьюрінга?
11. В чому суть УМТ?

ВПРАВИ

4.28.Перевірити, що МТ, яка працює за наведеною нижче ФС, залишає незмінним усяке парне число $2n$, а всяке непарне число $2n+1$ перетворює в число $2n$. Числа на стрічці записуються в алфавіті $\{I\}$.

Початкова конфігурація: $g_0 \Lambda I^n \Lambda$.

Функціональна схема:

Q / A	g_0	g_1	g_2
I	IПg_1	IПg_0	ΛН!
Λ	ΛН!	ΛЛg_2	

4.29.Побудувати МТ, яка перетворює кожне натуральне число n, записане в алфавіті $\{I\}$, в число $n+3$.

4.30.Скласти ФС машини Тьюрінга, яка потроює натуральні числа.

4.31.Побудувати МТ, яка б перетворювала кожне натуральне число n, записане в алфавіті $\{I\}$, в число $n+k$, де k - фіксоване натуральне число, записане в тому ж алфавіті.

4.32.Скласти ФС такої МТ, яка перетворює будь-яке натуральне число n в частку від ділення цього числа на два.

4.33.Побудувати МТ, яка додавала б раціональні дроби з однаковими знаменниками.

4.34.Побудувати МТ, яка перетворює будь-яке натуральне число n в остачу від ділення цього числа на п'ять.

4.35.Нехай в алфавіті $A=\{a,b\}$ задані слова P і Q. Скласти ФС МТ, яка слово $\alpha = P*Q$ переробляє в слово P.

4.36.Побудувати МТ, яка будь-яке слово $\alpha = P * Q * R$, де P, Q і R - слова в алфавіті $A = \{a, b, c\}$, перетворює в слово P.

4.37.Знайти ФС МТ, що перетворює кожне натуральне число *n* записане в алфавіті *{I}* в число:

$$\text{а) } \left[\frac{n+2}{3}\right]; \qquad \text{б) } \left[\frac{3n+1}{2}\right]$$

4.38.Побудувати МТ, що знаходить $|m-n|$, де n і m - натуральні числа, записані в алфавіті *{I}*.

4.39.Скласти ФС МТ, яка реалізує перетворення натуральних чисел, записаних в алфавіті *{I},* у запис цих самих чисел у двійковій системі числення.

4.40.Побудувати МТ, яка перетворює слово $1\underbrace{100...01}_{k \text{ разів}}$ в слово $\underbrace{100...01}_{k \text{ разів}}1$ (перенесення крайньої лівої одиниці в кінець слова).

4.5. МАШИНИ З НАТУРАЛЬНОЗНАЧНИМИ РЕГІСТРАМИ

Машина з натуральнозначними регістрами (скорочено МНР) є ідеалізованою моделлю комп'ютера. МНР містить, взагалі кажучи, нескінченну кількість регістрів, вмістом яких є натуральні числа. Регістри нумеруємо натуральними числами, починаючи з 0, позначаючи їх R_0, R_1, R_2 , ..., R_n,... Вміст регістру R_n позначаємо r_n.

Послідовність $(r_0\ ,\ r_1,\ ...,\ r_n\ ,\ ...)$ вмістів регістрів МНР назвемо *конфігурацією* МНР.

МНР може змінити вміст регістрів згідно виконуваної нею команди. Скінченний список команд утворює *програму* МНР. Команди програми послідовно нумеруємо натуральними числами, починаючи з 1.

Номер команди в програмі називатимемо *адресою* команди. МНР-програму з командами I_1 , I_2 ,..., $I_к$ будемо позначати I_1I_2,...$I_к$ Довжину (кількість команд) МНР-програми P позначимо |P|.

Команди МНР бувають наступних чотирьох типів.

Тип 1. Обнулення *n-го* регістру Z*(n)*: $r_n := 0$.

Тип 2. Збільшення вмісту *n-го* регістру на 1 S*(n)*: $r_n := r_n+1$.

Тип 3. Копіювання вмісту регістру *T(m,n)*: $r_n := r_m$ (при цьому r_m не змінюється).

Тип 4. Умовний перехід *J(m,n,q)*: *якщо* $r_n = r_m$ *, то перейти до виконання q-ї команди, інакше виконувати наступну за списком команду програми.*

Число q в команді $J(m,n,q)$ назвемо *адресою переходу.*

Команди типів 1-3 називають *арифметичними.* Після виконання арифметичної команди МНР повинна виконувати наступну за списком команду програми.

Виконання однієї команди МНР назвемо *кроком* МНР.

Зауважимо, що формальними моделями алгоритмів є саме МНР-програми, поняття МНР використовується для опису функціонування МНР-програм.

Виконання програми МНР починає, перебуваючи в деякій *початковій* конфігурації, з виконання 1-ї за списком команди. Наступна для виконання команда програми визначається так, як описано вище. Виконання програми завершується (програма зупиняється), якщо наступна для виконання команда відсутня (тобто номер наступної команди перевищує номер останньої команди програми). Конфігурація МНР в момент завершення виконання програми називається *фінальною,* вона визначає результат роботи МНР-програми над даною початковою конфігурацією.

МНР-програми, як моделі алгоритмів, є фінітними об'єктами, тому обмежимося розглядом скінченних конфігурацій. Конфігурацію вигляду $(a_0, a_1,..., a_n, 0, 0, ...)$, в якій $r_m= 0$ для всіх $m>n$, назвемо *скінченною.* Таку конфігурацію позначаємо $(a_0, a_1, ..., a_n)$. Зрозуміло, що якщо МНР-програма P починає роботу над скінченною початковою конфігурацією, то в процесі виконання P МНР перебуватиме тільки в скінченних конфігураціях.

МНР-програми P та Q назвемо *еквівалентними,* якщо при роботі над однаковими початковими конфігураціями вони або обидві зупиняються з однаковими фінальними конфігураціями, або обидві не зупиняються.

Нехай $n \in \{1, 2, 3....\}$. Кожну програму можна розглядати як алгоритм з множиною можливих початкових даних N^n. Застосування такого алгоритму до початкових даних $(x_1,..., x_n)$ полягає в наступному. У початковий момент числа $x_1,..., x_n$поміщаються відповідно в регістри $R_1,...,R_n$ при цьому в решті інших регістрів міститься 0. Потім виконуються команди даної програми, починаючи з першої. Один крок роботи алгоритму полягає у виконанні однієї команди. Послідовність кроків роботи алгоритму називається обчисленням. *Обчислення* завершується, коли в програмі немає команди, яку можна було б виконати. Це може відбутися, якщо

1) виконана остання команда програми, і ця команда була арифметичною;

2) при виконанні команди *J(m, n, q) виявилось,* що $r_m = r_n$, але q перевершує число команд в програмі;

3) при виконанні команди *J(m, n, q) виявилось*, що $r_m \neq r_n$, але це була остання команда програми.

Завершення роботи алгоритму завжди вважається результативним. Результатом роботи алгоритма є натуральне число, записане в регістрі R_0 у момент завершення обчислення. Таким чином, яким би не було $n \in \{1, 2, 3...\}$, кожна програма обчислює часткову n-місну числову функцію.

Означення. *Часткова функція f: $N^n \to N$ називається МНР-обчислювальною, якщо існує програма для МНР, яка обчислює цю функцію.*

Кожна МНР-програма обчислює *безліч* функцій, заданих на *N,* але, зафіксовуючи наперед *арність* функцій (тобто кількість компонент початкових конфігурацій), отримуємо, що кожна МНР-програма обчислює *єдину* функцію заданої арності.

Зауважимо, що кожну функцію, задану на *N*, можна трактувати як предикат, інтерпретуючи значення 1 та 0 як істинні значення "*T*" та "*F*" відповідно. В цьому випадку в ролі предикату виступає його характеристична функція.

Приклад 1. МНР-програма для функції $x+y$:

1) $J(1,2,5)$

2) $S(0)$

3) $S(2)$

4) $J(0,0,1)$

Приклад 2. МНР-програма для функції $x-y$:

1) $J(0,1,5)$

2) $S(1)$

3) $S(2)$

4) $J(0,0,1)$

5) $T(2,0)$

Приклад 3. МНР-програма для функції $[x/2]$:

1) $J(0,2,7)$

2) $S(2)$

3) $J(0,2,7)$

4) $S(2)$

5) $S(1)$

6) $J(0,0,1)$

7) $T(1,0)$

Приклад 4. МНР-програма для всюди невизначеної функції :

1) $J(0,0,1)$

Приклад 5. МНР-програма для функції $f:(x,y)=x-y$:

1) $J(0,1,7)$

2) $J(0,2,6)$

3) $S(1)$

4) $S(2)$

5) $J(0,0,1)$

6) $Z(2)$

7) $T(2,0)$

Приклад 6. МНР-програма для функції $f:(x,y)=x\bullet y$:

1) $J(3,1,9)$

2) $J(0,2,6)$

3) $S(2)$

4) $S(4)$

5) $J(0,0,2)$

6) $Z(2)$

7) $S(3)$

8) $J(0,0,1)$

9) $T(4,0)$

<u>Запитання для самоконтролю</u>

1. Що таке МНР?
2. Що тако конфігурація МНР?
3. Що таке програма МНР?
4. Опишіть команди МНР.
5. Як виконується програма МНР?
6. Дайте визначення еквівалентних МНР-програм.
7. Як визначається конкатенація стандартних МНР-програм?

8. Як визначається обчислюваність функції $f: N^n \to N$ за допомогою МНР-програми P?
9. Що таке МНР-обчислювана функція?

ВПРАВИ

4.41. Складсти МНР-програми для наступних функцій:

1) $f(x, y) = x - 2y$;
2) $f(x, y, z) = (x - y) + z$;
3) $f(x, y) = nsg(x + y)$;
4) $f(x, y) = sg(x \cdot y)$;
5) $f(x, y, z) = \max(x, y) + z$;
6) $f(x, y, z) = x - \min(y, z)$;
7) $f(x, y) = \max(x, 2y)$;
8) $f(x) = (x + 1) / 3$;
9) $f(x) = x!$;
10) $f(x, y) = x^y$.

4.42. Доведіть, що для кожної МНР-програми можна збудувати еквівалентну їй МНР-програму без команд *T(m,n)* (елімінація команд *T(m,n)*).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Алферова З.В. Теория алгоритмов. -М: Статистика, 1983.
2. Глушков В.М. Введение в кибернетику. -Киев: Изд.АНУССР, 1964.
3. Глушков В.М. Кибернетика. Вопросы теории и практики. -Москва: Наука, 1986.
4. Игошин В.И. Математическая логика и теория алгоритмов.- Изд-во Саратовського ун-та, 1991.
5. Євладенко В.М., Паращук С.Д. Методичні вказівки до практичних занять з математичної логіки.-Кіровоград:КДПІ, 1990.
6. Євладенко В.М., Халецька З.П., Нарадовий В.В. Математична логіка та теорія алгоритмів: Навчально- методичний посібник. - Кіровоград: Вид-во «КОД», 2009.
7. Калужнін Л.А, Королюк В.С. Алгоритми і математичні машини. -К.: Радянська школа, 1964.
8. Клини С. Математическая логика. М.: Мир, 1973.
9. Лісова Т.В. Математична логіка та теорія алгоритмів: [практикум].–Ніжин: Видавець ПП Лисенко М.М., 2011.- Частина 2.
10. Лиман Ф.М. Математична логіка і теорія алгоритмів.-Суми: Слобожанщина, 1998.
11. Мальцев А.И. Алгоритмы и рекурсивные функции. -М.: Наука, 1986.
12. Мендельсон Э. Введение в математическую логику. М.: Наука, 1971.
13. Марков А.А., Нагорный Н.М. Теория алгоритмов.-М.: Наука, 1984.
14. Олійник А.С., Сущанський В.І. Задачі з математичної логіки та теорії алгоритмів. Луганськ: ЛНПУ імені Тараса Шевченка, 2004.
15. Рамский Ю.С. Логічні основи інформатики: Навч посіб. – К.: НПУ імені М.П. Драгоманова, 2003.
16. Чуб О.Т., Коба В.І. Збірник вправ з курсу "Алгоритми і математичні машини". -К.: Вища школа, 1970.
17. Хромой Я.В. Математична логіка.- К.: Вища школа, 1983.

ПРЕДМЕТНИЙ ПОКАЗЧИК

Аксіоми числення L_1 51

Алгебра висловлень 6

Алгоритм 94

–уточнення 95

Алфавіт 6, 50, 67, 95

Атомарна формула 8, 64

Булева функція 23

Бінарна резольвента 79

Вивід 53

Вивідність із припущень 53

Висловлення 6

Гіпотеза Тюрінга 118

– Черча 111

Диз'юнкція 7, 60

Досконалий одночлен 18

Еквіваленція 7, 60

Заключна підстановка 97

Заперечення 7, 60

Імплікація 7, 60

Інтерпретація формули 69

Істинність висловлень 6

Квантор загальності 62

– існування 62

Клас булевих функцій 28

Комбінаційна схема 31

Кон'юнкція 7, 60

Кон'юнктивний одночлен 18

Конфігурація МТ 114

Категоричні судження 63

Літерал 7, 79

Логіка предикатів 59

Логічний наслідок 38, 43, 73

– критерій 40, 74

Машина Тьюрінга 113

Метасимволи 49, 50

Метатеорема дедукції 54

Метод резолюцій 79

МНР-програма 121

Множина істинності предиката 60

Нормальний алгоритм 95

– схема 96

Нормальні форми 18

– диз'юнктивна 18

– досконала 19

– зведена 76

– кон'юнктивна 18

– пренексна 76

– стандартна сколемівська 77

Область дії квантора 63

Обмежені квантори 88

Окремий випадок формули алгебри висловлень 75

Оператор підстановки 95

– розпізнавання 95

Операція суперпозиції 105, 106

– найменшого кореня 110

– примітивної рекурсії 106

Основні рівносильності 14, 75

Поліном Жегалкіна 26

Правила виводу 46, 55

Правило підстановки 51

Правило MP 51

Правило резолюцій 43, 80

Предикат 59

– тотожно істинний 59

– тотожно хибний 60

– виконуваний 60

– спростовний

Предметна змінна 67

– вільна 63

– зв'язана 63

– частково зв'язана 63

Принцип нормалізації 98

Рекурсивна функція 104

Рівносильні формули 14, 74

– ознака 14

РК-схема 34

Силогізми Арістотеля 74

Сума Жегалкіна 25

Стрілка Пірса 25

Схеми логічного слідування 40, 41

Таблиця істинності 7, 9

Терм 68

Теорема числення висловлень 52

Уніфікатор 81

Формула алгебри висловлень 8

– виконувана 10, 69

– нейтральна 10

– логіки предикатів 68

– логічно загальнозначуща 69

– суперечність 10, 77

– тавтологія 9,

– тотожно хибна 60

Функціональна повнота 26, 29

Функціональна схема МТ 115

Числення висловлень L_1 49

Штрих Шеффера 25