П. І. АВДЕЮК
О. В. ЗЕЛЕНСЬКИЙ

ЕЛЕМЕНТИ МАТЕМАТИЧНОЇ ЛОГІКИ ТА ТЕОРІЇ АЛГОРИТМІВ

Навчально-методичний посібник

П. І. АВДЕЮК, О. В. ЗЕЛЕНСЬКИЙ || ЕЛЕМЕНТИ МАТЕМАТИЧНОЇ ЛОГІКИ ТА ТЕОРІЇ АЛГОРИТМІВ

Кам'янець-Подільський національний університет
імені Івана Огієнка

П. І. АВДЕЮК,
О. В. ЗЕЛЕНСЬКИЙ

ЕЛЕМЕНТИ МАТЕМАТИЧНОЇ ЛОГІКИ ТА ТЕОРІЇ АЛГОРИТМІВ

Навчально-методичний посібник

Кам'янець-Подільський
2019

УДК 510.6:510.5(075.8)
ББК 22.12я73
А18

Рекомендувала вчена рада Кам'янець-Подільського національного університету імені Івана Огієнка, протокол № 2 від 28 лютого 2019 року.

Рецензенти:

І. М. Конет, доктор фіз.-мат. наук, професор;

М. І. Громяк, канд. фіз.-мат. наук, доцент;

І. В. Семенишина, канд. фіз.-мат. наук, доцент.

Авдеюк П. І., Зеленський О. В.

А18 **Елементи математичної логіки та теорії алгоритмів** : навчально-методичний посібник / П. І. Авдеюк, О. В. Зеленський. — Кам'янець-Подільський : Кам'янець-Подільський національний університет імені Івана Огієнка, 2019. — 160 с.

Посібник містить стислий виклад основ математичної логіки та теорії алгоритмів, а саме алгебру висловлень, числення висловлень, логіку предикатів, числення предикатів, рекурсивні функції, алгоритм як абстрактна машина. Для ілюстрації теоретичного матеріалу приведена достатня кількість вправ та прикладів, частика яких розв'язана в тексті, частина пропонується для самостійної роботи.

Книжка розрахована для початкового вивчення предмету, проте охоплює досить повний виклад матеріалу, передбаченого програмою для фізико-математичних факультетів університетів.

Для студентів математичних спеціальностей вищих навчальних закладів, вчителів математики та учнів старших класів спеціалізованих шкіл.

УДК 510.6:510.5(075.8)
ББК 22.12я73

ЗМІСТ

ПЕРЕДМОВА

Посібник написаний відповідно до програми курсу «Математична логіка і теорія алгоритмів» для фізико-математичних факультетів університетів і є логічним продовженням посібників «Елементи математичної логіки» та «Елементи теорії алгоритмів».

Фахівцю, зобов’язаному широко впроваджувати в навчальний процес обчислювальну техніку, необхідно не тільки вміти нею користуватися, але й знати ті теоретичні передумови, на яких заснована робота цієї техніки. Такою теоретичною базою є теорія алгоритмів.

Посібник складається з дев’яти розділів, в яких висвітлено основні питання математичної логіки та теорії алгоритмів, передбачені програмою.

Алгебра висловлень розглядається у першому розділі. У другому розділі — числення висловлень, логіку предикатів викладено в третьому розділі, числення предикатів розглядається в четвертому розділі. Поняття алгоритму на інтуїтивній основі аналізується у п'ятому розділі, також там встановлюється необхідність його уточнення. У шостому розділі описано рекурсивні функції. Вивченню одного з можливих уточнень поняття алгоритму — машинам Т’юринга відведено сьомий розділ. У восьмому розділі дано огляд іншого можливого уточнення поняття алгоритму — нормальних алгоритмів Маркова та сформульовано як пов'язані ці теорії між собою. У дев’ятому розділі наведено приклади алгоритмічно нерозв’язних задач.

ВСТУП

Математична логіка — розділ математики, в якому вивчаються математичні доведення та питання обґрунтування математики.

Математична логіка як самостійний розділ сучасної математики сформувався порівняно недавно — на межі XIX-XX століть.

Хоча логіка і є основою всіх інших наук, та притаманна їй, разом із фундаментальністю, властивість самоочевидності діяла розхолоджуюче на прагнення до скільки-небудь глибоких логічних досліджень аж до XIX століття, коли інтерес до логіки пожвавився під впливом відкриття неевклідових геометрій та намагання забезпечити строге обґрунтування математичного аналізу. Проте новий інтерес залишався не досить гострим до тих пір, поки на кінець століття математичний світ не був вражений відкриттям парадоксів, тобто міркувань, що приводять до суперечностей. Найбільш відомими парадоксами є такі:

1. (Б. Рассел, 1902 р.). Множина — сукупність будь-яких об'єктів. Множини можуть бути елементами множин.

Розглянемо множину A всіх множин X таких, що $X \notin X$. За означенням, якщо $A \in A$, то $A \notin A$ і навпаки, якщо $A \notin A$, то $A \in A$. В будь-якому випадку $A \in A$ і $A \notin A$, що суперечить закону виключеного третього.

2. (Д. Греллінг, 1908 р.). Прикметник називається автологічним, якщо властивість, яку він означає, притаманна йому самому. Прикметник називається гетерологічним, якщо властивість, яку він означає, йому самому не властива. Наприклад, «багатоскладний», «український» — автологічні, «односкладний», «російський», «їстівний» — гетерологічні.

Розглянемо прикметник «гетерологічний». Якщо цей прикметник гетерологічний, то він негетерологічний, якщо він негетерологічний, то він гетерологічний, тобто в будь-якому випадку прикметник «гетерологічний» є гетерологічним та негетерологічним одночасно.

Всі ці парадокси є істинними («справжніми») в тому розумінні, що вони не містять явних логічних помилок. У логічних парадоксах використовуються лише поняття теорії множин.

Виникнення парадоксів привернуло до питань основ математики велику увагу практично всіх провідних математиків того часу (Д. Гільберт, А. Пуанкаре, Г. Вейль).

Аналіз парадоксів призвів до різних планів їх усунення (зауважимо, що різноманітність підходів до основ математики залишається і сьогодні). Б. Рассел звернув увагу на те, що у всіх цих парадоксах має місце самовіднесення понять. Він запропонував забезпечити кожний об'єкт деяким невід'ємним цілим числом — "типом" цього об'єкта. Після цього твердження "X є елементом множини Y" може мати сенс тоді і тільки тоді, коли тип Y на одиницю більший типу X. Цей підхід систематично розвинений Расселом та Уайтхедом (1910-1913 рр.) в так звану теорію типів усуває відомі парадокси, проте він громіздкий в практичному застосуванні і має деякі інші недоліки. З іншого боку критика логічних парадоксів була спрямована на припущення, що містилося в них, про те, що для будь-якої властивості $P(x)$ існує відповідна множина елементів x, що мають властивість $P(x)$. Варто лише відкинути це допущення і логічні парадокси стають неможливими. Проте при цьому необхідно прийняти деякі нові аксіоми для того, щоб, спираючись на них, можна було довести існування таких множин, яких весь час потребують математики. Перша така аксіоматична теорія множин була побудована Цермело (1908 р.). Існують різні змішані теорії, що поєднують ті чи інші риси теорії типів і аксіоматичної теорії множин.

Більш глибоке тлумачення парадоксів було здійснено Л. Брауером та його інтуїціоністською школою (Л. Кронекер, Е. Борель, М. Лузін). Інтуїціоністи відмовляються визнавати універсальний характер деяких основних законів логіки, наприклад, закон виключеного третього (P або "не P"). Цей закон, стверджують вони, справджується лише для скінчених множин, але нема ніяких підстав поширювати його без жодних обмежень на всі множини. Згідно з їх думкою, ми тільки тоді можемо погодитись з твердженням про існування об'єкта, що має ту чи іншу властивість, коли ми володіємо методом побудови (або відшукання) такого об'єкта.

Апарат, що виник у математичній логіці, знайшов застосування в питаннях створення обчислювальних машин та програмного забезпечення до них. Логічне програмування — відносно новий перспективний напрямок сучасного програмування, що виник спочатку в рамках пошуків по створенню систем штучного інтелекту. Свою назву логічне програмування отримало від математичної логіки, що лежить в його основі. Зауважимо: із всієї математичної логіки використовується лише один із її розділів, а саме числення предикатів першого порядку. Основна мета створення логічного програмування — підвищення «інтелектуальності» комп'ютерів.

В зв'язку з цим виникає необхідність більш ширше і глибше вивчати математичну логіку. Адже саме з ідей цієї науки виникло точне означення поняття алгоритму, що дало можливість розв'язати багато питань, які без цього залишалися б в принципі нерозв'язними.

Цей невеликий курс «Елементи математичної логіки» є, по суті, лише «Вступом у математичну логіку» і відповідає потребам математичної та професійної підготовки вчителя математики та інформатики.

Розділ 1

АЛГЕБРА ВИСЛОВЛЕНЬ

1.1. ЛОГІЧНІ ОПЕРАЦІЇ. ФОРМУЛИ

Вчення про висловлення, яке називають алгеброю висловлень, є найпростішою із формальних логічних теорій. Під висловленням розуміємо розповідне речення, про яке можна сказати, що воно (його зміст) є істинним або хибним.

Саме ця властивість бути істинним чи хибним, є характеристичною для висловлення, як предмета вивчення логіки. Від усіх інших властивостей висловлень будемо абстрагуватися. При цьому вважаємо, що висловлення задовольняють закону виключеного третього, тобто кожне висловлення є або істинним або хибним, та закону виключення суперечності: жодне висловлення не може бути одночасно істинним і хибним.

Яке значення — "істинність" чи "хибність" властиве даному висловленню, залежить від відповідної реальності, якої стосується це висловлення. Будемо також розглядати висловлення істинність чи хибність яких на даний час ми точно встановити не можемо.

Наприклад, "2 — найменше просте число" є висловлення істинне, "3 > 5" — висловлення хибне, "Мадрид — столиця Франції" — висловлення хибне.

З урахуванням вищенаведеного, перше з цих висловлень може бути замінене символом "істинне", друге — "хибне", третє "хибне". Введемо символи, нейтральні щодо мови викладу. Позначимо значення "істинне" та "хибне" відповідно через "1" та "0". Звичайно "1" і "0" тут не є назвами чисел, а тільки символами значень істинності.

Вказаний підхід математичною мовою можна сформулювати так: на множині висловлень задано функцію, яка набуває точно два значення: "істинне" або "хибне". Цю функцію називають *функцією істинності*.

Для позначення висловлень (з урахуванням висловлень, істинність яких нам невідома) введемо символи $a, b, c, ..., p, q, ...$. Їх називають *пропозиційними змінними*. Значення функції істинності для даного значення змінної p позначатимемо $|p|$.

В алгебрі висловлень вводять *логічні операції* над висловленнями.

Назва	Позначення	Тип	Інші позначення
заперечення	$\rceil$	*унарний*	$\overline{\ }$, *not, ні*
кон'юнкція	$\wedge$	*бінарний*	*&, and, і*
диз'юнкція	$\vee$	*бінарний*	*or, або*
імплікація	$\Rightarrow$	*бінарний*	$\supset$, $\rightarrow$
еквіваленція	$\Leftrightarrow$	*бінарний*	$\sim$, $\leftrightarrow$

ОЗНАЧЕННЯ 1.1.1. *Операції, за допомогою яких із вихідних висловлень отримують нові висловлення, називаються логічними операціями. Позначатимемо:* $\rceil$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$.

Пропозиційні змінні, операції, дужки називають *алфавітом алгебри висловлень.*

ОЗНАЧЕННЯ 1.1.2. *Вираз побудований із пропозиційних змінних за допомогою логічних операцій називається пропозиційною формулою або формулою алгебри висловлень, при виконанні таких умов:*

1) всі пропозиційні змінні — пропозиційні формули;

2) якщо A, B — пропозиційні формули, то $(\rceil A)$, $(\rceil B)$, $(A\wedge B)$, $(A\vee B)$, $(A\Rightarrow B)$, $(A\Leftrightarrow B)$ *— пропозиційні формули;*

3) тільки ті вирази є пропозиційними формулами для яких це випливає із п.1 і п.2.

За допомогою цього означення легко перевірити, чи є дане слово (тобто дана послідовність логічних символів) формулою.

Для такої перевірки треба:

1) замінити кожну пропозиційну змінну змінною a.

2) замінити кожну формулу виду $(\rceil a)$, $(a \wedge a)$, $(a \vee a)$, $(a \Rightarrow a)$, $(a \Leftrightarrow a)$ літерою a, повторюючи цю заміну доти, доки маємо хоча б одну формулу вказаного виду.

Якщо в результаті одержимо a, то дане слово — формула. В іншому випадку дане слово не є формулою.

Приклад 1.1.1. Чи є формулою слово $(a \Rightarrow ((\rceil b)\vee c))$?

Розв'язання.

1. З $(a \Rightarrow ((\rceil b)\vee c))$ одержимо, підставивши замість кожної змінної a: $(a \Rightarrow ((\rceil a)\vee a))$.

2. Замінимо $(\rceil a)$ через a, маємо $(a \Rightarrow (a \vee a))$.

3. Замінивши $(a \vee a)$ на a, одержимо $(a \Rightarrow a)$.

4. Замінивши $(a\Rightarrow a)$ на a, одержимо a. Слово $(a\Rightarrow((\rceil b)\vee c))$ формула.

Приклад 1.1.2. Чи є формулою слово $(a \vee ((b \Rightarrow (\rceil c))\wedge b)$?

Розв'язання.

1. З $(a \vee ((b \Rightarrow (\rceil c))\wedge b)$ заміною всіх змінних на a одержимо $(a\vee((a \Rightarrow (\rceil a))\wedge a)$.

2. Заміною $(\rceil a)$ на a одержимо $(a \vee (a \Rightarrow a)\wedge a)$.

3. Заміною $(a \Rightarrow a)$ на a одержимо $(a \vee a \wedge a)$. Одержаний вираз не містить жодної формули виду $(\rceil a)$, $(a \vee a)$, $(a \wedge a)$, $(a \Rightarrow a)$, $(a \Leftrightarrow a)$ і не є формулою (a). Отже, слово $(a \vee ((b \Rightarrow (\rceil c))\wedge b))$ не є формулою.

Дужки визначають однозначний порядок виконання операцій.

ОЗНАЧЕННЯ 1.1.3. *Логічна операція називається головною операцією, якщо вона у формулі виконується останньою.*

$$((a\vee(в\Rightarrow c)) \Leftrightarrow ((c\vee a) \wedge (a\Rightarrow((\rceil в) \wedge c)))). \qquad (1.1.1)$$

Тут головною операцією є еквіваленція — $\Leftrightarrow$.

Кожну формулу алгебри висловлень можна позначити великою літерою латинської абетки ("назвати", "дати ім'я").

Наприклад формулу (1.1.1) можна позначити A.

Але при такому запису не враховується той факт, що у формулу входять різні пропозиційні змінні. Для більшої точності слідувало б перерахувати всі пропозиційні змінні, наприклад, у порядку $a, b, \dots z, a_1, b_1, \dots, a_2, b_2, \dots$ і перепозначити їх — $p_1, p_2, p_3, \dots$ Тоді скорочений запис формули (1.1.1) буде мати своїми аргументами p_1, p_2, p_3, тобто $A(p_1, p_2, p_3)$.

Нехай задана формула алгебри висловлень $A(p_1, \dots, p_n)$, де пропозиційні змінні $p_1, \dots, p_n$ можуть приймати довільним чином значення істинності 0 або 1.

ОЗНАЧЕННЯ 1.1.4. *Кожний розподіл значень істинності всіх пропозиційних змінних $p_1, \dots, p_n$, називають n-значним набором $(p_1, \dots, p_n)$.*

Кожному набору $(p_1, \dots, p_n)$ відповідає точно визначене єдине значення істинності формули $A(p_1, \dots, p_n)$, яке можна легко обчислити.

Значення істинності формули може бути різним при різних наборах. Таким чином кожна формула визначає деяку функцію, аргументами якої є пропозиційні змінні.

Для того, щоб розглянути залежність істинностного значення $A(p_1, \dots, p_n)$ від усіх можливих розподілів істинностних значень змінних $p_1, \dots, p_n$, необхідно впорядкувати множину всіх наборів, щоб не пропустити жодного. Оскільки аргументи і функції приймають тільки два значення, то така функція цілком може бути описана скінченою таблицею.

Наведемо таблиці елементарних функцій: $\rceil a$, $a \wedge b$, $a \vee b$, $a \Rightarrow b$, $a \Leftrightarrow b$.

a	b	$\rceil a$	$a \wedge b$	$a \vee b$	$a \Rightarrow b$	$a \Leftrightarrow b$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

В цій таблиці кожній операції відповідає стовпець, кожному набору — рядок.

Вищенаведені табличні означення логічних операцій рівносильні таким означенням:

ОЗНАЧЕННЯ 1.1.5. *Заперечення ($\rceil a$) істинне тільки тоді, коли a — хибне. ($\rceil a$) — хибне тоді, коли a істинне.*

ОЗНАЧЕННЯ 1.1.6. *Кон'юнкція ($a \wedge b$) істинна тоді і тільки тоді, коли обидва компоненти a і b є одночасно істинними. В решті випадків кон'юнкція хибна.*

ОЗНАЧЕННЯ 1.1.7. *Диз'юнкція ($a \vee b$) хибна тоді і тільки тоді, коли обидві компоненти a і b є одночасно хибними. Якщо принаймні одна з її компонент є істинною, то диз'юнкція істинна.*

ОЗНАЧЕННЯ 1.1.8. *Імплікація ($a \Rightarrow b$) хибна тоді і тільки тоді, коли її антецедент a — істинний, консеквент b — хибний. В решті випадків імплікація істинна.*

ОЗНАЧЕННЯ 1.1.9. *Еквіваленція ($a \Leftrightarrow b$) істинна тоді і тільки тоді коли a та b набувають однакових значень істинності, в інших випадках еквіваленція хибна.*

Взагалі, для кожної формули алгебри висловлень можна побудувати відповідну таблицю істинності. Але при трьох пропозиційних змінних у формулі таблиця матиме 8 рядків, при п'яти пропозиційних змінних — 32 рядки, при десяти пропозиційних змінних — 1024 рядки, взагалі, при k пропозиційних змінних таблиця матиме 2^k рядки.

Приклад 1.1.3. Складемо таблицю істинності формули (1.1.1):

a	b	c	$b \Rightarrow c$	$a \vee (b \Rightarrow c)$	$c \vee a$	$\rceil b$	$\rceil b \wedge c$	$b \Rightarrow ((\rceil b) \wedge c$	$(c \vee a) \wedge (a \Rightarrow ((\rceil b) \wedge c))$	A
0	0	0	1	1	0	1	0	1	0	0
0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	0	0	1	0	1
0	1	1	1	1	1	0	0	1	1	1
1	0	0	1	1	1	1	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0

Таблиця істинності для формули алгебри висловлень задає певну функцію істинності. Але не слід змішувати формулу алгебри висловлень з функцією істинності, яку ця формула визначає. Формула є "назвою" відповідної функції істинності. Функція істинності може зображатися різними формулами.

З метою мінімізації запису формул алгебри висловлень будемо застосовувати такі правила скорочення дужок:

1. Зовнішні дужки опускаються.

2. Введемо ранг логічних операцій: $\rceil$, $\wedge$, $\vee$,$\Rightarrow$,$\Leftrightarrow$. Це означає, що логічна операція, яка стоїть лівіше у наведеній послідовності зв'язує пропозиційні літери сильніше, ніж та, яка відносно неї стоїть правіше в цій послідовності. Отже сильніше за всіх зв'язує заперечення, слабше за нього кон'юнкція, але вона зв'язує сильніше ніж решта операцій. Найслабше зв'язує еквіваленція.

Наприклад, скорочений запис формули (1.1) за цими правилами має вигляд:

$$a \vee (b \Rightarrow c) \Leftrightarrow (c \vee a) \wedge (a \Rightarrow \rceil b \wedge c).$$

Вправи

1. З'ясувати, чи є дана послідовність формулою:

a) $(a \wedge b)\ c\ \rceil b;$

b) $(a \wedge b) \Rightarrow \rceil b;$

c) $((a \wedge \rceil b) \Rightarrow b\ (a \Rightarrow b);$

d) $(((\rceil a) \Rightarrow b) \Rightarrow \rceil(a \Rightarrow \rceil b)).$

2. Розставити дужки так, щоб отримати формулу з таких послідовностей:

a) $a \wedge b \Rightarrow a \Leftrightarrow \rceil b \vee \rceil a;$

b) $a \wedge \rceil b \Rightarrow b;$

c) $c \Rightarrow a \Rightarrow b$

d) $a \wedge b \vee c \Leftrightarrow \rceil b;$

e) $a \vee \rceil b \Rightarrow c \wedge d$

3. Скласти таблиці істинності для формул:

a) $(a \Rightarrow b) \vee ((a \Rightarrow \rceil b) \wedge \rceil a);$

b) $((a \Rightarrow b) \Rightarrow a) \wedge b;$

c) $(a \wedge (b \Rightarrow \rceil b)) \wedge ((\rceil b \Rightarrow a) \Rightarrow b);$

d) $((a \Rightarrow \rceil b) \Rightarrow b) \vee (a \Rightarrow b);$

e) $(a \wedge (b \wedge (\rceil a \Rightarrow \rceil b));$

f) $((a \Rightarrow b) \vee b) \wedge b;$

g) $((a \Rightarrow \rceil b) \Rightarrow b) \Rightarrow (\rceil a \Rightarrow b).$

1.2. ТАВТОЛОГІЇ

Важливу роль відіграють ті формули алгебри висловлень, які істинні незалежно від того, які значення приймають пропозиційні змінні, що входять в них. Такі формули називаються тавтологіями.

ОЗНАЧЕННЯ 1.2.1. *Формула алгебри висловлень називається тавтологією, якщо функція істинності, яка визначається цією формулою, тотожно рівна 1.*

Тавтології називаються також тотожно істинними формулами або законами алгебри висловлень. Якщо формула A є тавтологією, то позначатимемо це так $\models A$.

Наведемо деякі найважливіші тавтології.

1. $a \Leftrightarrow \neg\neg a$ — закон подвійного заперечення;
2. $a \vee \neg a$ — закон виключеного третього;
3. $\neg(a \wedge \neg a)$ — закон виключення суперечності;
4. $a \Leftrightarrow a$ — закон тотожності;
5. $a \wedge a \Leftrightarrow a,\ a \vee a \Leftrightarrow a$ — закони ідемпотентності;
6. $a \wedge b \Rightarrow a,\ a \wedge b \Rightarrow b$ — закон спрощення;
7. $a \Rightarrow a \vee b,\ b \Rightarrow a \vee b$ — закон спрощення;
8. $(a \Rightarrow b) \Leftrightarrow (\neg b \Rightarrow \neg a)$ — закон контрапозиції;
9. $(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$ — закон силогізму;
10. $(a \Leftrightarrow b) \Leftrightarrow (\neg a \Leftrightarrow \neg b)$ — закон протилежності;
11. $a \Rightarrow (b \Rightarrow a)$ — істина з чого завгодно;
12. $\neg a \Rightarrow (a \Rightarrow b)$ — з хибного що завгодно;
13. $a \wedge (a \Rightarrow b) \Rightarrow b$ — закон modus ponens;
14. $(a \Rightarrow b) \wedge \neg b \Rightarrow \neg a$ — закон modus tollens;
15. $\neg(a \wedge b) \Leftrightarrow \neg a \vee \neg b$ —закон де Моргана;
16. $\neg(a \vee b) \Leftrightarrow \neg a \wedge \neg b$ — закон де Моргана;
17. $a \wedge (a \vee b) \Leftrightarrow a$ — закон поглинання;
18. $a \vee a \wedge b \Leftrightarrow a$ — закон поглинання;
19. $a \Leftrightarrow (b \Rightarrow a \wedge b)$.

Доведення цих законів пропонуємо виконати самостійно за допомогою таблиць істинності.

Часто для доведення того, що дана формула є тавтологією, використовують метод відшукання *контрприкладу*.

Приклад 1.2.1. Перевіримо, чи є тавтологією формула

$$(a \Rightarrow b) \Rightarrow (a \vee c \Rightarrow b \vee c). \qquad (1.2.1)$$

Розв'язання. Припустимо, що формула (1.2.1) не є тавтологією. Тоді знайдеться хоча б один набір (a_0, b_0, c_0) на якому истинісне значення формули (1.2.1) дорівнюватиме 0. Оскільки головна операція у формулі (1.2.1) — імплікація, то, за означенням імплікації $|(1.2.1)| = 0$ тільки тоді, коли

$$|a_0 \Rightarrow b_0| = 1 \qquad (1.2.2)$$

та

$$|a_0 \vee c_0 \Rightarrow b_0 \vee c_0| = 0. \qquad (1.2.3)$$

З (1.2.3) дістаємо

$$|a_0 \vee c_0| = 1, \qquad (1.2.4)$$

$$|b_0 \vee c_0| = 0, \qquad (1.2.5)$$

за означенням імплікації.

Далі, з формули (1.2.5), за означенням диз'юнкції, отримаємо

$$|b_0| = 0, \qquad (1.2.6)$$

$$|c_0| = 0. \qquad (1.2.7)$$

Підставивши значення (1.2.7) у формулу (1.2.4), дістаємо

$$|a_o| = 1. \qquad (1.2.8)$$

А підставивши значення (1.2.8) і (1.2.6) у формулу $a \Rightarrow b$ отримаємо $|a_0 \Rightarrow b_0| = |1 \Rightarrow 0| = 0$, що суперечить (1.2.2), отже наше припущення хибне, тобто формула (1.2.1) — тавтологія.

Важливим методом одержання нових тавтологій є підстановка у дану формулу.

ТЕОРЕМА 1.2.1. *Якщо $A(p_1, ..., p_n)$ — тавтологія, то формула A^*, одержана із формули A підстановкою довільної формули алгебри висловлень B замість кожного входження літери p_i в A, також є тавтологією.*

Доведення. $|A(p_1, ..., p_{i-1}, 1, p_{i+1}, ..., p_n)| = 1$ та

$$|A(p_1, ..., p_{i-1}, 0, p_{i+1}, ..., p_n)| = 1,$$

оскільки A — тавтологія. Але B не може приймати інших значень крім 0 і 1 , тобто $|A^*|$ тотожно дорівнює 1.

Приклад 1.2.2. Відомо, що $\models (a \Rightarrow b) \Leftrightarrow (\bar{a} \Rightarrow \bar{b}) = A(a, b)$.

Підставимо в $A(a,b)$ замість a формулу $a \Rightarrow \bar{b}$. В результаті дістанемо формулу

$$((a \Rightarrow \bar{b}) \Rightarrow b) \Leftrightarrow (\overline{(a \Rightarrow \bar{b})} \Rightarrow \bar{b}),$$

яка за теоремою 1.2.1 також є тавтологією.

ТЕОРЕМА 1.2.2. *Якщо $\models A$ та $\models A \Rightarrow B$, то $\models B$.*

Доведення. Припустимо, що B не є тавтологією, тобто існує набір значень істинності пропозиційних змінних на якому $|B| = 0$, тоді на цьому наборі $|A \Rightarrow B| = 0$, що суперечить умові.

Приклад 1.2.3. Підстановкою у тавтологію $\bar{a} \Rightarrow (a \Rightarrow b)$ замість a формули $a \wedge \bar{a}$ дістаємо $\models \overline{(a \wedge \bar{a})} \Rightarrow ((a \wedge \bar{a}) \Rightarrow b)$. Оскільки $\models \neg(a\wedge b)$ – закон виключення суперечності, то за теоремою 1.2.2 отримаємо $\models(a\wedge \neg a)\Rightarrow b$.

ОЗНАЧЕННЯ 1.2.2. *Формула алгебри висловлень, яка приймає значення істинності 0, на кожному наборі значень пропозиційних змінних називається суперечністю. Такі формули позначатимемо 0.*

Тотожно істинні формули позначатимемо 1.

ОЗНАЧЕННЯ 1.2.3. *Формула алгебри висловлень, яка не є ні тавтологією ні суперечністю називається нейтральною.*

ОЗНАЧЕННЯ 1.2.4. *Формула алгебри висловлень, яка не є суперечністю називається виконуваною.*

Отже формули алгебри висловлень розподіляються так:

<table>
<tr><td rowspan="2">Суперечності</td><td>Нейтральні</td><td>Тавтології</td></tr>
<tr><td colspan="2">Виконувані</td></tr>
</table>

Вправи

1. Скласти таблиці істинності для формул та вказати, які з формул є виконуваними, які – тавтологіями, які – суперечностями:

a) $(a\Rightarrow b)\Rightarrow((a\Rightarrow \neg b)\Rightarrow \neg a)$;
b) $((a\Rightarrow b)\Rightarrow a)\Rightarrow b$;
c) $(a\wedge(b\vee \neg b))\wedge((\neg b\Rightarrow a)\vee b)$;
d) $((a\wedge \neg b)\Rightarrow b)\Rightarrow(a\Rightarrow b)$;
e) $(a\wedge(b\wedge(\neg a\vee \neg b))$;
f) $((a\Rightarrow b)\Rightarrow b)\Rightarrow b$;
g) $((a\vee \neg b)\Rightarrow b)\wedge(\neg a\vee b)$.

2. Довести, що ці формули виконувані:

a) $\neg(a\Rightarrow \neg a)$;
b) $(a\Rightarrow b)\Rightarrow(b\Rightarrow a)$;
c) $(b\Rightarrow a\wedge c)\wedge \neg(a\vee c\Rightarrow b)$;
d) $\neg((a\Leftrightarrow \neg b)\vee c)\wedge b$;
e) $a\wedge b\Rightarrow(c\vee b\Rightarrow b\wedge \neg b)$.

3. Довести чи спростувати, що ці формули є тавтологіями:

a) $(a\Rightarrow b)\wedge(b\Rightarrow c)\Rightarrow(a\Rightarrow c)$;
b) $(a\Rightarrow b)\Rightarrow((a\Rightarrow(b\Rightarrow c))\Rightarrow(a\Rightarrow c))$;
c) $(a\Rightarrow c)\Rightarrow((b\Rightarrow c))\Rightarrow(a\vee b\Rightarrow c))$;
d) $a\Rightarrow b\wedge c\Leftrightarrow(a\Rightarrow b)\wedge(a\Rightarrow c)$;
e) $((a\Rightarrow b)\Rightarrow(c\Rightarrow d))\Rightarrow((a\Rightarrow c)\Rightarrow(b\Rightarrow d))$;
f) $(a\Rightarrow b)\wedge(a\Rightarrow d)\wedge(a\wedge c)\Rightarrow(a\Rightarrow b\wedge c\wedge d)$;
g) $(a\Rightarrow(b\Rightarrow c))((c\Rightarrow d)\Rightarrow(a \Rightarrow(b\Rightarrow d)))$.

4. Визначити, які з формул алгебри висловлень є: а) тавтологіями, б) суперечностями, в) нейтральними:

a) $(a\vee b\Rightarrow c)\Rightarrow(a\Rightarrow c)\vee(b\Rightarrow c)$;
b) $(a\wedge b\Rightarrow c)\Rightarrow(a\Rightarrow c)\wedge(b\Rightarrow c)$;
c) $(\neg a\vee b)\wedge(\neg b\vee c)\wedge a\wedge \neg c)$;
d) $(a\Rightarrow b)\wedge(c\Rightarrow d)\Rightarrow(a\wedge c\Rightarrow b\wedge d)$;
e) $(a\wedge c\Rightarrow b\wedge d)\Rightarrow (a\Rightarrow b)\wedge(c\Rightarrow d)$;
f) $((a\Leftrightarrow b)\Rightarrow(c\Leftrightarrow d))\Rightarrow(a\vee c\Rightarrow b\vee d)$;
g) $(a\Rightarrow b)\Rightarrow((c\Leftrightarrow d)\Rightarrow(a\vee c\Leftrightarrow b\vee d))$.

5. Методом відшукання контрприкладу встановити, що ці формули – тавтології:

a) $(a\Rightarrow b)\wedge(c\Rightarrow d)\Rightarrow(a\vee c\Rightarrow b\vee d)$;
b) $a\vee b\Rightarrow((a\Rightarrow c)\wedge(c\Rightarrow d)\Rightarrow c\vee d)$;
c) $(a\Rightarrow(b\Rightarrow c))\Rightarrow((a\Rightarrow b)\Rightarrow(a\Rightarrow c))$;
d) $(a\Rightarrow(b\Rightarrow c))\Rightarrow((d\Rightarrow a)\Rightarrow(d\Rightarrow(b\Rightarrow c))$;
e) $(a\Rightarrow(b\Rightarrow c))\Rightarrow((d\Rightarrow b)\Rightarrow(a\Rightarrow(d\Rightarrow c))$.

1.3. РІВНОСИЛЬНІСТЬ ФОРМУЛ АЛГЕБРИ ВИСЛОВЛЕНЬ

ОЗНАЧЕННЯ 1.3.1. *Формули алгебри висловлень $A(p_1, p_2, ..., p_n)$ та $B(p_1, p_2, ..., p_n)$ називаються рівносильними, якщо на кожному наборі значень пропозиційних змінних, ці формули приймають однакові значення істинності. (Якщо вони мають одну і ту ж функцію істинності).*

Якщо формули A та B рівносильні, то позначатимемо це так:

$$A \equiv B.$$

З означення рівносильності випливає, що для будь-яких формул алгебри висловлень A, B, C мають місце властивості:

1) $A \equiv A$ — рефлексивність;
2) якщо $A \equiv B$, то $B \equiv A$ — симетричність;
3) якщо $A \equiv B$ і $B \equiv C$, то $A \equiv C$ — транзитивність.

Тобто відношення рівносильності є відношенням еквівалентності.

Наведемо приклади основних рівносильних формул:

1. $a \wedge b \equiv b \wedge a$ — комутативний закон;
2. $a \vee b \equiv b \vee a$ — комутативний закон;
3. $a \wedge (b \wedge c) \equiv (a \wedge b) \wedge c$ — асоціативний закон;
4. $a \vee (b \vee c) \equiv (a \vee b) \vee c$ — асоціативний закон;

5. $a\wedge (b\vee c) \equiv a\wedge b\vee a\wedge c$ — перший дистрибутивний закон;
6. $a\vee b \wedge c \equiv (a\vee b) \wedge (a\vee c)$ — другий дистрибутивний закон;
7. $a\wedge a \equiv a$ — закон ідемпотентності;
8. $a\vee a \equiv a$ — закон ідемпотентності;
9. $a\vee a\wedge b\equiv a$ — закон поглинання;
10. $a\wedge(a\vee b) \equiv a$ — закон поглинання;
11. $\rceil(a\wedge b) \equiv \rceil a \vee \rceil b$ — закон де-Моргана;
12. $\rceil(a\vee b) \equiv \rceil a \wedge \rceil b$ — закон де-Моргана;
13. $a\Rightarrow b \equiv \rceil b \Rightarrow \rceil a$ — закон контрапозиції;
14. $a\Rightarrow b \equiv \rceil a \vee b$;
15. $a \Leftrightarrow b \equiv (a \Rightarrow b) \wedge (b \Rightarrow a)$;
16. $\rceil\rceil a \equiv a$ — закон подвійного заперечення;
17. $a \vee \rceil a = 1$ — закон виключеного третього;
18. $a \wedge \rceil a \equiv 0$ — закон суперечності.

Для довільної формули A мають місце закони сталих:

19. $A \wedge 0 \equiv 0$;
20. $A \wedge 1 \equiv A$;
21. $A \vee 0 \equiv A$;
22. $A \vee 1 \equiv 1$.

Доведемо рівносильність (5):

$$a \wedge (b \vee c) \equiv a \wedge b \vee a \wedge c.$$

Нехай $|a\wedge(b\vee c)| = 0$. Тоді $|a| = 0$ або $|b\vee c| =0$ за означенням кон'юнкції. Якщо $|a| =0$, то $|a\wedge b| = 0$ і $|a\wedge c| =0$.

Отже $|a\wedge b\vee a\wedge c| =0$. Якщо $|b| =0$ і $|c| =0$, то $|a\wedge b| = 0$ і $|a\wedge c| = 0$. Отже $|a\wedge b\vee a\wedge c| = 0$. Тобто в обох випадках права частина має те саме значення, що й ліва частина.

Нехай тепер $|a \wedge (b \vee c)| =1$. Тоді $|a| =1$ і $|b \vee c| = 1$.

Тоді

$$|b| = 1, \qquad (*)$$

або

$$|c| = 1. \qquad (**)$$

Якщо матимемо рівність (*), то $|a\wedge b| = 1$, а якщо (**), то $|a\wedge c| = 1$. Отже за означенням диз'юнкції права частина рівносильності (5) має значення 1, тобто і в цьому випадку збігається із значенням лівої частини.

Доведення рівносильностей 1-22 можна здійснити також за допомогою таблиць істинності. Пропонуємо зробити це самостійно.

Спираючись на означення рівносильності і тавтології, легко зробити висновок, що кожну тавтологію в якій головна операція еквіваленція можна замінити на рівносильність двох відповідних формул.

З рівносильністю формул зв'язане поняття рівносильності висловлень.

ОЗНАЧЕННЯ 1.3.2. *Два висловлення називаються рівносильними якщо формули які зображають їх логічну структуру є рівносильними.*

Якщо має місце рівносильність $A \equiv B$, то має місце тавтологія $A \Leftrightarrow B$ і навпаки. Тобто можна будь-яку рівносильність записати як тавтологію, з допомогою теореми підстановки можна замінити пропозиційні літери довільними формулами алгебри висловлень. Одержимо тавтологію. Зробивши зворотній перехід від тавтології з головною операцією еквіваленція до рівносильності одержимо узагальнені рівносильності.

ТЕОРЕМА 1.3.1. *Нехай $A \equiv B$, A_1 — підформула A, $A_1 \equiv A_2$. Позначимо A^* — результат заміни у формулі A підформулу A_1 на A_2. Тоді $A^* \equiv B$. Аналогічно, якщо у формулі B замінити підформулу на рівносильну. Якщо у рівносильності $A \equiv B$ замінити будь-яку підформулу A або B на рівносильну їй, то рівносильність не порушиться.*

Доведення безпосередньо випливає із способу побудови таблиці істинності.

Відмінність між заміною і підстановкою:

1) підстановка здійснюється тільки замість пропозиційної змінної, заміна — замість підформули;

2) підставляти можна будь-яку формулу, заміна тільки на рівносильну формулу;

3) підстановка здійснюється обов'язково в усіх місцях входження пропозиційної літери, заміна підформули — замість одної входження її у формулу.

Вправи

1. Довести рівносильності:

a) $a \wedge (a \vee c) \wedge (b \vee c) \equiv a \wedge b \vee a \wedge c;$
b) $(a \vee b) \wedge (b \vee c) \wedge (c \vee a) \equiv a \wedge b \vee b \wedge c \vee c \wedge a;$
c) $(a \vee b) \wedge (b \vee c) \wedge (c \vee d) \equiv a \wedge b \vee b \wedge c \vee c \wedge d;$
d) $(a \vee b \vee c) \wedge (b \vee c \vee d) \wedge (c \vee d \vee a) \equiv a \wedge b \vee b \wedge d \vee d \wedge a \vee c;$
e) $(a \vee b) \wedge (\neg b \vee \neg a) \vee b \wedge a \equiv a \vee b;$
f) $(a \Leftrightarrow b) \Rightarrow a \wedge c \equiv a \wedge c \vee a \wedge \neg b \vee b \wedge \neg a;$
g) $(a \vee b \vee c) \wedge (b \vee c \vee d) \wedge (c \vee d \vee a) \equiv a \vee ((b \vee c \wedge d) \wedge (d \vee c);$
h) $a \wedge (b \wedge c). \equiv (a \wedge b) \wedge c$;
i) $a \vee (b \vee c) \equiv (a \vee c) \vee c$;
j) $a \wedge (b \vee c) \equiv a \wedge b \vee a \wedge c$;
k) $a \vee b \wedge c \equiv (a \vee b) \wedge (a \vee c).$

1.4. НОРМАЛЬНІ ФОРМИ. ПРОБЛЕМА ВИРІШЕННЯ

Задача: вказати єдиний спосіб (алгоритм), який дозволяє для кожної формули A алгебри висловлень з'ясувати за скінченну кількість кроків, чи є формула A тавтологією, чи ні. Якщо ми маємо такий спосіб, то ми одночасно отримаємо також і спосіб визначити, чи буде дана формула виконуваною чи ні. Дійсно, ми можемо для довільної формули розв'язати питання: $\rceil A$ — тавтологія, чи ні. Якщо $\rceil A$ — тавтологія, то A — суперечність , якщо $\rceil A$ не є тавтологією, то A не є суперечністю, отже буде виконуваною формулою.

Поставлена задача носить назву проблеми вирішення. Ця проблема ставиться не тільки для алгебри висловлень, але і для інших логічних систем. Для алгебри висловлень ця проблема легко розв'язується за допомогою таблиць істинності.

Існує інший спосіб, який ґрунтується на зведенні формул до так званої "нормальної форми".

ОЗНАЧЕННЯ 1.4.1. *Кон'юнктом (диз'юнктом) називається кон'юнкція (диз'юнкція) пропозиційних змінних та їх заперечень (при цьому допускається повторення змінних). Число пропозиційних змінних у кон'юнкті (диз'юнкті) називають його рангом.*

ОЗНАЧЕННЯ 1.4.2. *Диз'юнкція (кон'юнкція) n кон'юнктів (диз'юнктів) (n = 1, 2, 3, ...) називається n-членною диз'юнктивною (кон'юнктивною) нормальною формою. Скорочений запис відповідно днф і кнф.*

ОЗНАЧЕННЯ 1.4.3. *Днф (кнф), яка рівносильна даній формулі алгебри висловлень A, називається днфA (кнфA).*

Доведемо, що для кожної формули алгебри висловлень A існує кнфA і днфA. Для цього задамо скінчену послідовність кроків, внаслідок яких, виходячи з A, дістанемо кнфA або днфA (конструктивне доведення).

1) Користуючись у разі необхідності рівносильностями $A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A)$, $A \Rightarrow B \equiv \rceil A \vee B$, замінюємо A рівносильною формулою A_1, в якій немає жодного символу логічної операції, відмінного від $\rceil$, $\wedge$, $\vee$.

2) Застосовуючи закони де-Моргана і закон подвійного заперечення, дістанемо формулу A_2, в якій кожний символ заперечення стосується тільки тієї пропозиційної змінної, яка безпосередньо за ним слідує, причому $A_2 \equiv A$.

3) Щоб дістати днфA (кнфA), досить, у разі необхідності, застосувати до A_2 перший (другий) дистрибутивний закон.

Приклад 1.4.1. Знайдемо днф і кнф формули $a\wedge b \equiv \neg(a\vee b)$. З метою обґрунтування кожної ланки ланцюга рівносильностей, тобто кожного кроку доведення, ми відразу записуватимемо цей ланцюг вертикально, записуючи справа від кожної формули її обґрунтування (в дужках) та нумеруючи кожен крок:

1. $a\wedge b \Leftrightarrow \neg(a\vee b)$;
2. $(a\wedge b\Rightarrow \neg(a\vee b))\wedge(\neg(a\vee b)\Rightarrow a\wedge b)$ (рівносильність 16 до ф. 1);
3. $(\neg(a\wedge b)\vee\neg(a\vee b))\wedge(\neg\neg(a\vee b)\vee a\wedge b)$ (рівносильність 14 до ф. 2);
4. $(\neg a\vee\neg b)\vee\neg(a\vee b))\wedge(\neg\neg(a\vee b)\vee a\wedge b)$ (рівносильність 11 до ф. 3);
5. $((\neg a\vee\neg b)\vee v\neg a\wedge\neg b)\wedge(\neg\neg(a\vee b)\vee a\wedge b)$ (рівносильність 12 до ф. 4);
6. $((\neg a\vee\neg b)\vee a\neg b))\wedge((a\vee b)\vee a\wedge b)$ (рівносильність 15 до ф. 5).

Для зведення до днф застосовуємо перший дистрибутивний закон.

7. $((\neg a\vee\neg b)\vee\neg a\wedge\neg b)\wedge(a\vee b))\vee(((\neg a\vee\neg b)\vee\neg a\wedge\neg b)\wedge a\wedge b$ (6, рівносильність 5);
8. $((\neg a\vee\neg b)\vee\neg a\wedge\neg b)\wedge a\vee((\neg a\vee b)\vee\neg a\wedge\neg b)\wedge b\vee(\neg a\vee\neg b)\wedge a\wedge b\vee\neg a\wedge\neg b\wedge a\wedge b$ (7, рівносильність 5);
9. $\neg a\wedge a\vee\neg b\wedge a\vee\neg a\wedge\neg b\wedge a\vee\neg a\wedge b\vee\neg b\wedge b\vee\neg a\wedge\neg b\wedge b\vee\neg a\wedge a\wedge b\vee\neg b\wedge a\wedge b\vee\vee\neg a\wedge\neg b\wedge a\wedge b$ (8, рівносильність 5).

Формула 9 є диз'юнкцією кон'юнктів, отже вона є днф формули 1.

Для зведення вихідної формули до кнф застосуємо другий дистрибутивний закон до формули 6.

7*. $(\neg a\vee\neg b\vee\neg a)\wedge(\neg a\vee\neg b\vee\neg b)\wedge(a\vee b\vee a)\wedge(a\vee b\vee b)$ (6, рівносильність 6).

Формула 7* є кон'юнкцією диз'юнктів отже вона є кнф формули 1.

Нормальні форми дають змогу розв'язати проблему вирішення.

ТЕОРЕМА 1.4.1. *Формула $A(p_1, ..., p_n)$ алгебри висловлень є тавтологією тоді і тільки тоді, коли кожен диз'юнкт, що входить до кнфA, містить хоч одну пропозиційну змінну p_i ($i = 1, ..., n$) разом з її запереченням.*

Достатність. Дано, що кожен диз'юнкт для деякого i має вигляд: $...\vee p_i \vee ... \neg p_i \vee ...$. Завдяки комутативності й асоціативності диз'юнкції можна замінити розглядуваний диз'юнкт на рівносильний йому, який матиме вигляд $(...(p_i\vee\neg p_i)...)$. Оскільки $|p_i\vee\neg p_i| = 1$ та для довільної формули A: $|A\vee 1| = 1$, то отримали, що істинісне значення даного кон'юнктивного члена дорівнює 1, незалежно від істинісних значень змінних $p_1, ..., p_n$. Цей результат стосується кожного кон'юнктивного члена кнфA. Отже, |кнфA| = 1 на кожному наборі ($p_1, ..., p_n$). Через те, що кнф$A \equiv A$, то $\models A(p_1, ..., p_n)$.

Необхідність. Дано, що $A(p_i, ..., p_n)$ — тавтологія. Припустимо, що хоч один диз'юнкт у кнфA не містить жодної пропозиційної змінної $p_1,..., p_n$ разом з її запереченням. Позначимо його D. Надамо кожній змінній p_i, яка входить в D під знаком заперечення, істинісне значення 1, а тим пропозиційним змінним p_k, які входять в D без знака заперечення, — істинісне значення 0. При цьому розподілі істинісних значень D =0. Беручи до уваги рівносильність $A\wedge 0 = 0$ для довільної формули A, дістаємо $|$кнф$A| = |A| = 0$ при даному розподілі істинісних значень $p_1, ..., p_n$. Це суперечить тому, що формула $A(p_{1i}, ..., p_n)$ є тавтологією. Таким чином, зроблене припущення — неправильне. Отже, якщо A — тавтологія, то кожен диз'юнкт у кнфA містить хоч одну пропозиційну змінну $p_1,..., p_n$ разом з її запереченням.

Приклад 1.4.2. Чи є тавтологією формула $(\neg a\Rightarrow b)\wedge(\neg a\Rightarrow \neg b)\Rightarrow a$?

1. $(\neg a \Rightarrow b)\wedge(\neg a\Rightarrow \neg b)\Rightarrow a$;
2. $\neg((\neg\neg a\vee b)\wedge(\neg\neg a\vee \neg b))\vee a$ рівносильність 14 до ф.1;
3. $\neg((a\vee b)\wedge(a\vee \neg b))\vee a$ рівносильність 15 до ф. 2;
4. $\neg(a\vee b\wedge \neg b)\vee a$ рівносильність 6 до ф. 3;
5. $(\neg a\wedge(\neg b\vee \neg\neg b))\vee a$ рівносильність 11,12 до ф. 4;
6. $\neg a\wedge(\neg b\vee b)\vee a$ рівносильність 15 до ф. 5;
7. $(\neg a\vee a)\wedge(\neg b\vee b\vee a)$ рівносильність 6 до ф. 6.

Остання формула є кнф, яка складається з двох кон'юнктивних членів (двох диз'юнктів). Перший диз'юнкт містить пропозиційну змінну a разом з її запереченням, другий — пропозиційну змінну b разом з її запереченням. Отже, за теоремою 1.4.1, дана формула — тавтологія.

Аналогічно тавтологією є формула $(a \Rightarrow b) \vee (b\Rightarrow a)$.

Оскільки вона рівносильна формулі $(\neg a \vee b) \vee (\neg b \vee a)$ яка складається з одного кон'юнктивного члена (з одного диз'юнкта). У цей диз'юнкт входить пропозиційна змінна a разом з її запереченням.

Зауважимо, що для кожної формули алгебри висловлень існують істотно різні кнф і днф (звичайно, всі вони рівносильні між собою).

Виділимо тепер з класу нормальних форм для даної формули A алгебри висловлень певний підклас, члени якого однозначно визначаються за даною формулою A (з точністю до порядку запису). Введемо

ОЗНАЧЕННЯ 1.4.4. *Кон'юнкт (диз'юнкт) називається правильним, якщо кожна пропозиційна змінна входить до нього не більше ніж один раз, включаючи також входження змінної під знаком заперечення.*

ОЗНАЧЕННЯ 1.4.5. *Кон'юнкт (диз'юнкт) називається повним щодо пропозиційних змінних $p_1, ..., p_n$, якщо кожна з цих змінних входить до нього або під знаком запереченим, або без нього.*

ОЗНАЧЕННЯ 1.4.6. *Досконалою диз'юнктивною (кон'юнктивною) нормальною формою щодо пропозиційних змінних $p_1, ..., p_n$ називається днф (кнф), в якій всі кон'юнкти (диз'юнкти) правильні і повні щодо даного набору пропозиційних змінних.*

Досконалу диз'юнктивну (кон'юнктивну) нормальну форму позначатимемо так: Дднф (Дкнф).

Щоб звести днф, в якій жодний диз'юнктивний член не повторюється, до Ддиф, треба:

1) Кожен неправильний кон'юнкт, користуючись комутативним і асоціативним законами, законом суперечності і рівносильностями (7), (19), (20) перетворити в правильний.

2) До неповного кон'юнкта, який не містить, наприклад, пропозиційної змінної p_i приєднати кон'юнктивно диз'юнкт $p_i \vee \neg p_i$, істинісне значення якого тотожно дорівнює 1, і розкрити дужки за першим дистрибутивним законом. У разі необхідності цей процес повторюється доти, поки кожен неповний кон'юнкт не перетвориться у повний.

3) Якщо в остаточній днф є однакові диз'юнктивні члени, то залишити тільки один з них.

Щоб звести кнф, жодний кон'юнктивний член якої не повторюється, до Дкнф, треба:

1) Кожен неправильний диз'юнкт, застосовуючи комутативний і асоціативний закони, закон виключеного третього та рівносильності (8), (21), (22), перетворити у правильний.

2) До неповного диз'юнкта, який не містить, наприклад, пропозиційної змінної p_i, приєднати диз'юнктивно кон'юнкт $p_i \wedge \neg p_i$, істинісне значення якого тотожно дорівнює 0, і розкрити дужки за другим дистрибутивним законом. У разі необхідності цей процес повторюється доти, поки кожен неповний диз'юнкт не перетвориться в повний.

Якщо в остаточній кнф є однакові кон'юнктивні члени, то залишити тільки один з них.

Приклад 1.4.3. Перетворити днф у рівносильну їй Ддиф:

1. $a \wedge b \wedge c \wedge \neg a \vee a \wedge \neg b \vee \neg a \wedge \neg b \wedge c \wedge \neg b \vee a \wedge c \wedge a$;
2. $(a \wedge \neg a) \wedge b \wedge c \vee a \wedge \neg b \vee a \wedge (\neg b \wedge \neg b) \wedge c \vee (a \wedge a) \wedge c$ (комутативний і асоціативний закони кон'юнкції до формули 1);
3. $0 \wedge b \wedge c \vee a \wedge \neg b \vee \neg b \wedge \neg b \wedge c \vee a \wedge c$(закон суперечності і рівносильність до ф.2);
4. $0 \vee a \wedge \neg b \vee \neg a \wedge b \wedge c \vee a \wedge c$ (рівносильність 19, 3);
5. $a \wedge \neg b \vee \neg a \wedge \neg b \wedge c \vee a \wedge c$ (рівносильність 21, 4);

6. $a\wedge \neg b \wedge (c\vee \neg c) \vee \neg a \wedge \neg b \wedge c \vee a \wedge c\wedge (b\wedge \neg b)$ (рівносильності 17, 20 до формули 5);
7. $a \wedge \neg b \wedge c \vee a \wedge \neg b; c \vee \neg a \wedge \neg b \wedge \neg b \wedge c \vee a \wedge c\wedge b \vee a \wedge c \wedge \neg b$ (рівносильність 5 до 6);
8. $a\wedge \neg b\wedge c\vee a\wedge \neg b\wedge \neg c\vee \neg b\wedge c\wedge a\wedge b\wedge c\vee a\wedge \neg b\wedge c$ (комутативний закон кон'юнкції до ф.7);
9. $(a\wedge \neg b\wedge c \vee a \neg b\wedge c) \vee a\wedge \neg b\wedge \neg c \vee \neg a\wedge \neg b\wedge c \vee a \wedge \neg b \wedge c$ (комутативний і асоціативний закони кон'юнкції до формули 8);
10. $a\wedge \neg b\wedge c\vee a\wedge \neg b \wedge \neg c \vee \neg a \neg b\wedge c\vee a\wedge b\wedge c$ (рівносильність 8 до формули 9).

У формулі 10 кожний диз'юнктивний член є правильним і повним кон'юнктом і жоден кон'юнкт не повторюється. Отже ця формула і є шуканою Дднф.

Приклад 1.4.4. Перетворити кнф у рівносильну їй Дкнф:

1. $(a \vee b \vee \neg c) \wedge (a \vee b \vee \neg c \vee \neg b) \wedge (a \vee b) \wedge (b \vee c)$;
2. $(a \vee b \vee \neg c) \wedge (a \vee (b \vee \neg b) \vee \neg c) \wedge (a \vee b) \wedge (b\vee c)$ (комутативний і асоціативний закон її диз'юнкції до формули 1),
3. $(a \vee b \vee \neg c) \wedge (a \vee 1 \wedge \neg c)\wedge (a \vee b) \wedge (b \vee c)$ (рівносильність 17, до формули 2);
4. $(a\vee b\vee \neg c)\wedge 1\wedge (a\vee b)\wedge (b\vee c)$ (рівносильність 22, до формули 3);
5. $(a\vee b\vee \neg c)\wedge (a\vee b)\wedge (b\vee c)$ (рівносильність 20, до формули 4);
6. $(a\vee b\vee b)\wedge (a\vee b\wedge c\wedge \neg c)\wedge (a\wedge b\wedge b\vee c)$ (рівносильності 18 і 21 до формули 5);
7. $(a\vee b\vee \neg c)\wedge (a\vee b\vee c)\wedge (a\vee b\vee \neg c)\wedge (a\vee b\vee c)\wedge (\neg a\vee b\vee c)$ (рівносильність 6 до формули 6),
8. $((a\vee b\vee \neg c)\wedge (a\vee b\vee \neg c))\wedge ((a\vee b\vee c)\wedge (a\vee b\vee c))\vee (\neg a\vee b\vee c)$ (комутативність і асоціативність кон'юнкції до формули 7);
9. $(a\vee b\vee \neg c)\wedge (a\vee b\vee c)\wedge (\neg a\vee b\vee c)$ (рівносильність 7, до формули 8).

У формулі 9 кожний кон'юнктивний член є правильним і повним диз'юнктом і жоден диз'юнкт не повторюється. Отже ця формула і є шуканою Дкнф.

ТЕОРЕМА 1.4.2. *Для того щоб формула $A(p_1, ..., p_n)$ алгебри висловлень була тавтологією, необхідно і досить, щоб Ддснф A містила 2^n кон'юнктів (тобто всі можливі правильні і повні кон'юнкти, що містять змінні $p_1, ..., p_n$).*

Доведення. Необхідність. Дано, що $|A| = 1$ при всіх розподілах істинісних значень $p_1, ..., p_n$. Припустимо, що ДдснфA не містить 2^n кон'юнктів. Тоді знайдеться такий кон'юнкт, що складений з $p_1, ..., p_n$, який не входить у ДдснфA. Позначимо його через K. Надамо кожній змінній, яка входить в K без знака заперечення, значення 1, а кожній змінній, що входить в K із знаком заперечення, — значення 0. При такому розподілі істинісних значень $p_1, \ldots, p_n$ матимемо $|K| = 1$. Тоді всі кон'юнкти, що входять у ДднифA, відрізняючись від K хоч одним логічним множником, матимуть значення 0 при даному розподілі істинісних значень $p_1, \ldots, p_n$. Звідси $|\text{Ддниф}A| = 0$ на цьому наборі, що суперечить умові теореми. Необхідність доведена.

Достатність. За умовою ДднифA містить 2^n кон'юнктів. Зазначимо, що число всіх різних наборів $p_1, \ldots, p_n$ теж становить 2^n. Розглянемо довільний набір значень $p_1, \ldots, p_n$, тобто довільну упорядковану n-ку з нулів та одиниць. Поставимо їй у взаємно-однозначну відповідність кон'юнкт K_1, складений з $p_1, \ldots, p_i, \ldots, p_n$ за таким правилом:

— одиниці на i-му місці в n-ці відповідає p_i в K_1,

— нулю на i-му місті в n-ці відповідає $\rceil p_i$ в K_1.

Очевидно, на розглядуваному наборі значень $p_1, \ldots, p_n$ маємо $|K_1| = 1$. Отже, $|\text{Ддниф}A| = 1$, бо K_1 входить як диз'юнктивний член в ДднифA. Беручи до уваги довільність розглянутого набору, дійдемо висновку, що $|\text{Ддниф}A| = 1$ при кожному розподілі істинісних значень $p_1, \ldots, p_n$. Оскільки $A \equiv$ ДднифA, то A тавтологія. Достатність доведено.

Вправи

1. Рівносильними перетвореннями привести кожну з вказаних формул до днф:

a) $\rceil(a \vee c) \wedge (a \Rightarrow b)$;
b) $(a \Leftrightarrow b) \wedge \rceil(c \Rightarrow d)$;
c) $(a \vee b \wedge \rceil c(\wedge(a \vee c)$;
d) $((a \Rightarrow b) \Rightarrow (c \Rightarrow \rceil a)) \Rightarrow (\rceil b \Rightarrow \rceil c)$;
e) $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow \rceil c) \Rightarrow (a \Rightarrow \rceil b))$.

2. Рівносильними перетвореннями привести кожну з формул завдання 1 до кнф.

3. Рівносильними перетвореннями привести кожну з вказаних формул до ддниф:

a) $(\rceil a \vee c) \wedge (a \Rightarrow b)$;
b) $a \wedge b \vee b \wedge c$;
c) $a \vee b \wedge c$;
d) $a \wedge b \vee c \wedge d$;
e) $(a \wedge \rceil b \vee c) \wedge (\rceil a \vee c)$;
f) $(a \vee c) \wedge (a \vee b) \vee \rceil b \wedge (c \vee \rceil b)$;
g) $a \vee b \vee c$.

4. Рівносильними перетвореннями привести кожну з вказаних формул до дкнф:

a) $\rceil a \wedge c \vee c \wedge b$;
b) $(a \vee c) \wedge d$;
c) $(\rceil a \vee b) \wedge (a \vee c)$;
d) $\rceil a \wedge b \vee c \wedge d$;
e) $a \wedge b \wedge c \vee d$;
f) $a \wedge b \vee b \wedge c \vee c \wedge d$;
g) $a \wedge b \vee c$.

1.5. ФУНКЦІЇ АЛГЕБРИ ВИСЛОВЛЕНЬ

ОЗНАЧЕННЯ 1.5.1. *Булевою називається функція, значення якої, як і значення всіх її аргументів, належать двоелементній множині.*

Якщо взяти двоелементну множину $X = \{0, 1\}$, то булева функція від n аргументів (n-місна) є відображенням множини $\{0, 1\}^n$ на $\{0, 1\}$. Отже, функції істинності алгебри висловлень є булевими функціями. Булеву n-місну функцію $f(x_1, ..., x_n)$ ($x_i \in \{0, 1\}$) можна задати:

1) таблицею, яка визначає яке значення f відповідає кожному n-значному набору значень аргументів $x_1,..., x_n$;

2) формулою, яка визначає операції, котрі необхідно виконати над значенням аргументів $x_1,..., x_n$, щоб отримати значення функції;

3) словесно:

$$f(x_1,..., x_n) = \begin{cases} 1, \text{ при } |x_i| = 1, \text{ де } i = 2k\ (k \in N); \\ 0, \text{ в інших випадках.} \end{cases}$$

Оскільки таблиця, що визначає функцію n змінних містить 2^n рядків (число кортежів з 2 по n), а в кожному рядку може стояти одне з двох значень: 0 або 1, то всього існує 2^{2^n} логічних функцій від n змінних. При цьому може виявитись, що від деяких змінних функція фактично не залежить. Такі змінні називаються фіктивними.

Приклад 1.5.1. Для функцій $f_i(a, b, c)$, ($i = 1, 2, 3$), вказати фіктивні змінні.

№	a	b	c	$f_1(a, b, c)$	$f_2(a, b, c)$	$f_3(a, b, c)$
1	0	0	0	1	1	0
2	0	0	1	1	0	0
3	0	1	0	1	1	0
4	0	1	1	1	0	1
5	1	0	0	0	0	1
6	1	0	1	0	1	1
7	1	1	0	1	0	1
8	1	1	1	1	1	1

Розв’язання. Функція $f_1(a, b, c)$ набуває однакових значень в рядках 1-2, 3-4, 5-6, 7-8, які відрізняються тільки значенням c, отже, ця функція від c не залежить. В той же час в рядках 1-5, які відрізняються тільки значенням a, і в рядках 5-7, які відрізняються тільки значенням b, функція набуває різних значень, отже, від a та b функція залежить, а c є фіктивною змінною.

Функція $f_2(a, b, c)$ набуває різних значень в рядках 1-2, отже, залежить від c; в рядках 1-5, отже залежить від a. В той же час, в рядках 1-3, 2-4, 5-7, 6-8 вона набуває однакових значень, отже, b є фіктивною змінною. Функція $f_3(a, b, c)$ набуває різних значень в рядках 1-5, отже, залежить від a; 2-4, отже, залежить від b; 3-4, отже, залежить від c. Ця функція не містить фіктивних змінних.

Число всіх наборів $(x_1, ..., x_n)$ дорівнює $N = 2^n$. А кількість n-місних булевих функцій дорівнює $A_2^N = 2^N = 2^{2^n}$.

При $n = 1$ (одномісні функції) маємо $2^2 = 4$ ($f_1=0$, $f_2=1$, $f_3=x$, $f_4 = \neg x$).

При $n=2$ маємо $2^{2^2} = 2^4 = 16$ — двомісних булевих функцій, які задаються таблицею:

x	y	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
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

При $n = 3$ маємо $2^{2^3} = 2^8 = 256$ тримісних функцій.

ТЕОРЕМА 1.5.1. *Кожна булева функція зображується формулою алгебри висловлень, яка містить символи не більш ніж трьох пропозиційних зв'язок —* $\neg, \wedge, \vee$.

Доведення. Нехай $f(x_1, ..., x_n)$ — довільна булева функція, тобто кожному (з 2^n можливих) набору значень аргументів відповідає одне з двох значень 0 чи 1. Пронумеруємо всі набори та j-му набору ($1 \le j \le 2^n$) поставимо у відповідність кон'юнкт:

$$K_j = \mathop{\wedge}_{i=1}^{n} x_i^* \quad \text{де} \quad x_i^* = \begin{cases} x_i, & \text{якщо } |x_i| = 1; \\ \neg x_i, & \text{якщо } |x_i| = 0. \end{cases}$$

Вважаємо, що хоча б на одному наборі $|f| = 1$, інакше $f \equiv 0$, тоді $f = x_i \wedge \neg x_i$.

Розглянемо деякий набір, на якому $|f| = 1$. Нехай це буде r-ий набір, тоді $|K_r| = 1$. На решті наборів $|K_r| = 0$ через те, що всі останні набори відрізняються від r-ого хоча б одним значенням x.

Побудуємо диз'юнкцію D всіх K_j, які відповідають всім тим наборам на яких f приймає значення 1. Ця диз'юнкція і буде шуканою формулою. Дійсно, для кожного набору, на якому $|f| = 1$, в D знайдеться $K : |K| = 1$ на цьому наборі, тоді $|D| = 1$. На кожному наборі, де $|f| = 0$, всі $|K| = 0$ і тому $|D| = 0$ на цьому наборі. Таким чином D зображує функцію f, причому в D не входять інші зв'язки, крім $\neg, \wedge, \vee$.

Отже, будь-яка булева функція $f \neq 0$ зображається у вигляді Ддиф.

Приклад 1.5.2. Булева функція задана таблицею істинності

№	x	y	z	$f(x, y\ z)$
1	0	0	0	1
2	0	0	1	0
3	0	1	0	1
4	0	1	1	1
5	1	0	0	0
6	1	0	1	0
7	1	1	0	1
8	1	1	1	0

Написати Ддиф, яка реалізує $f(x, y, z)$.

Розв'язання. Виділимо набори на яких f набуває значення 1, і утворимо для кожного набору відповідний кон'юнкт K_j:

$$K_1 = \rceil x \wedge \rceil y \wedge \rceil z;\ K_3 = \rceil x \wedge y \wedge \rceil z;\ K_4 = \rceil x \wedge y \wedge z;\ K_7 = x \wedge y \wedge \rceil z.$$

Кожен з цих кон'юнктів набуває значення "1" тільки на одному, відповідному йому наборі, на всіх інших наборах він дорівнює 0.

Диз'юнкція D цих кон'юнктів:

$$D = \rceil x \wedge \rceil y\ \rceil z \vee \rceil x \wedge y \wedge \rceil z \vee \rceil x \wedge z \wedge y \vee x \wedge y \wedge \rceil z.$$

$|D| = 1$ на першому, третьому, четвертому, сьомому наборах, де значення f дорівнює 1, і $|D| = 0$ на всіх інших наборах.

Отже, побудована формула D і є шуканою Ддиф.

Приклад 1.5.3. Записати формули для функцій $f_1(a, b, c)$, $f_2(a, b, c)$, $f_3(a, b, c)$ з прикладу 1.5.1 і спростити ці формули.

Розв'язання. Функція $f_1(a, b, c)$ набуває значення істинності, рівного 1, в рядках 1, 2, 3, 4, 7, 8. Випишемо кон'юнкти, які відповідають цим наборам. Якщо значення істинності змінної дорівнює 1, беремо саму змінну, якщо значення істинності змінної дорівнює 0, беремо заперечення змінної. Отже, маємо:

— набору 0,0,0 відповідає кон'юнкт $\rceil a \wedge \rceil b \wedge \rceil c$;
— набору 0,0,1 відповідає кон'юнкт $\rceil a \wedge \rceil b \wedge c$;
— набору 0,1,0 відповідає кон'юнкт $\rceil a \wedge b \wedge \rceil c$;
— набору 0,1,1 відповідає кон'юнкт $\rceil a \wedge b \wedge c$;
— набору 1,1,0 відповідає кон'юнкт $a \wedge b \wedge \rceil c$;
— набору 1,1,1 відповідає кон'юнкт $a \wedge b \wedge c$.

Отже, $f_1(a, b, c) = \rceil a \wedge \rceil b \wedge \rceil c \vee \rceil a \wedge \rceil b \wedge c \vee \rceil a \wedge b \wedge \rceil c \vee \rceil a \wedge b \wedge c \vee a \wedge b \wedge \rceil c \vee a \wedge b \wedge c$.

Легко переконатись, що одержана формула справді визначає функцію $f_1(a, b, c)$, тобто що при кожному наборі значень змінних формула має те ж значення істинності, що і функція

$f(a, b, c)$. Спростимо цю формулу. Згрупуємо кон'юнкти по два, скориставшись асоціативністю диз'юнкції. Маємо

$$(\rceil a\wedge\rceil b\wedge\rceil c\vee\rceil a\wedge\rceil b\wedge c)\vee(\rceil a\wedge b\wedge\rceil c\vee\rceil a\wedge b\wedge c)\vee(a\wedge b\wedge\rceil c\vee a\wedge b\wedge c).$$

Винесемо за дужки, скориставшись дистрибутивністю кон'юнкції відносно диз'юнкції:

$$\rceil a\wedge\rceil b\wedge(\rceil c\vee c)\vee\rceil a\wedge b\wedge(\rceil c\vee c)\vee a\wedge b\wedge(\rceil c\vee c).$$

За законом виключеного третього $(\rceil c\vee c)$ є тотожно істинною формулою. Її можна замінити числом 1. За законом сталих кон'юнктивний член 1 можна опустити, маємо $\rceil a\wedge\rceil b\vee\rceil a\wedge b \vee a\wedge b$.

Далі перетворюємо: $(\rceil a\wedge\rceil b\vee\rceil a\wedge b)\vee(\rceil a\wedge b\vee a\wedge b) = =\rceil a\wedge(b\vee b)\vee b\wedge(\rceil a\vee a) =\rceil a\vee b=a\Rightarrow b$.

Аналогічно складаються і спрощуються формули, що виражають функції $f_2(a, b, c)$ і $f_3(a, b, c)$:

$$f_2(a, b, c) = \rceil a\wedge\rceil b\wedge\rceil c\vee\rceil a\wedge b\wedge\rceil c\vee a\wedge\rceil b\wedge c\vee a\wedge b\wedge b =$$
$$= (\rceil a\wedge\rceil b\wedge\rceil c\vee\rceil a\wedge b\wedge\rceil c)\vee(a\wedge\rceil b\wedge c\vee a\wedge b\wedge c) =$$
$$= \rceil a\wedge\rceil c\wedge(\rceil b\vee b)\vee a\wedge c\wedge(\rceil b\vee b)=\rceil a\wedge\rceil c\vee a\wedge c=a\Leftrightarrow c.$$
$$f_3(a, b, c)=\rceil a\wedge b\wedge c\vee a\wedge\rceil b\wedge\rceil c\vee a\wedge\rceil b\wedge c\vee a\wedge b\wedge\rceil c\vee a\wedge b\wedge c =$$
$$= b\wedge c\wedge(\rceil a\vee a)\vee a\wedge\rceil b\wedge(\rceil c\vee c)\vee a\wedge b\wedge(\rceil c\vee c) =$$
$$= b\wedge c\vee(a\wedge\rceil b\vee a\wedge b) = b\wedge c\vee a.$$

ТЕОРЕМА 1.5.2. *Зображення булевої функції, відмінної від тотожного нуля, Ддиф є єдиним (з точністю до порядку членів).*

Доведення. Оскільки кожна із змінних $x_1, ..., x_n$ обов'язково зустрічається в кожному повному і правильному кон'юнкті, або із знаком заперечення або без нього, то число всіх таких кон'юнктів складених з $x_1, ..., x_n$ дорівнює числу розміщень з повторенням з 2 елементів по n, тобто 2^n. Число всіх Ддиф, складених з цих кон'юнктів, дорівнює числу всіх підмножин множини кон'юнктів, зменшеному на одиницю, тобто $2^{2^n}-1$ (віднімання одиниці обумовлено тим, що множина всіх підмножин включає порожню множину, а поняття порожньої Ддиф не введене).

Але число всіх булевих функцій від змінних $x_1, ..., x_n$ відмінних від тотожного нуля також дорівнює числу $2^{2^n}-1$.

Припустимо, що для деякої $f^*(x_1, ..., x_n)$ існує більше ніж одна Ддиф, що зображує f^*. Тоді хоча б для однієї $f(x_1, ..., x_n)$, відмінної від тотожного нуля, не існує Ддиф, яка її зображає, що суперечить попередній теоремі.

Отже, для кожної булевої функції існує не більше ніж одна Ддиф, яка її зображає, а з урахуванням попередньої теореми — точно одна Ддиф.

ТЕОРЕМА 1.5.3. *Кожна булева функція, відмінна від тотожної одиниці, зображається Дкнф яка є єдиною (з точністю до порядку).*

Доведення. Нехай f — довільна булева функція тотожно не рівна одиниці. Тоді $\neg f \neq 0$ і зображується Ддиф, яку позначимо D. Зображенням $\neg\neg f$ є формула алгебри висловлень $\neg D$. За законами подвійного заперечення і де-Моргана, $\neg\neg f \equiv f$, $\neg D \equiv$ Дкнф. Теорема доведена.

Приклад 1.5.4. Булева функція $f(x, y, z)$ задана таблицею.

x	y	z	$f(x, y, z)$	$\neg f(x, y, z)$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

Написати Дкнф, яка реалізує $f(x, y, z)$.

Розв'язання. Маючи на увазі доведення попередньої теореми, виписуємо поруч із стовпчиком значень f аналогічний стовпчик для $\neg f$. Ддиф, що реалізує $\neg f$, матиме вигляд:

$$D = \neg x \wedge \neg y \wedge \neg z \vee \neg x \wedge y \wedge z \vee y \wedge z.$$

Тоді $\neg\neg f$, яка збігається з f, зобразиться формулою $\neg D$:

$$\neg(\neg x \wedge \neg y \wedge \neg z \vee \neg x \wedge y \wedge z \vee x \wedge y \wedge z) = (x \vee y \vee z) \wedge (x \vee \neg y \vee \neg z) \wedge (\neg x \vee \neg y \vee \neg z).$$

Тут використано закони де-Моргана і закон подвійного заперечення. Оскільки остання формула є Дкнф, то вона є шуканою.

ОЗНАЧЕННЯ 1.5.2. *Система операцій алгебри висловлень називається функціонально повною, якщо кожну булеву функцію можна зобразити формулою, яка не містить інших символів логічних операцій, крім тих, що входять в цю систему.*

Система операцій $\{\neg, \wedge, \vee\}$ є функціонально повною. Це твердження безпосередньо випливає з теореми 1.5.1.

ТЕОРЕМА 1.5.4. *Системи операцій 1)* $\{\neg, \wedge\}$; *2)* $\{\neg, \vee\}$; *3)* $\{\neg, \Rightarrow\}$; *4)*$\{\Rightarrow, 0\}$ *є функціонально повними.*

Доведення. 1. Оскільки система $\{\neg, \wedge, \vee\}$ є функціонально повною, то досить показати, що диз'юнкцію можна виразити формулою алгебри висловлень, яка не містить інших символів операцій, крім $\neg$, $\wedge$. Це встановлюється рівносильністю $p \vee q \equiv \neg(\neg p \wedge \neg q)$.

2. Доведемо аналогічно 1), тільки використовуємо рівносильність р $\wedge\, q \equiv \neg(\neg p \vee \neg q)$.

3. Скористаємося рівносильністю $p \vee q \equiv \rceil p \Rightarrow q$ та повнотою системи 2).

4. Скористаємося рівносильністю $\rceil p \equiv p \Rightarrow 0$ і повнотою системи 3).

Приклад 1.5.5. Довести, що система, яка складається з однієї операції $a/b \equiv \rceil(a \wedge b)$, — функціонально повна.

Розв'язання. Покажемо, що через дану операцію (/ — штрих Шеффера) можна виразити кон'юнкцію, диз'юнкцію і заперечення.

Легко бачити, що $a/a \equiv \rceil a$.

Отже, заперечення можна виразити через дану операцію. Тоді

$$a \wedge b \equiv \rceil (a/b) \equiv (a/b)/(a/b),$$

$$a \vee b \equiv \rceil ((\rceil a) \wedge (\rceil b)) \equiv (\rceil a)/(\rceil b) \equiv (a/a)/(b/b).$$

Дана система, що складається з операції /, повна.

Приклад 1.5.6. Чи повна система, що складається з однієї операції $\Rightarrow$?

Розв'язання. Перевіримо, чи можна виразити через імплікацію решту операцій, а саме кон'юнкцію, диз'юнкцію і заперечення. Оскільки $a \wedge b \equiv \rceil(a \Rightarrow \rceil b)$, $a \vee b \equiv \rceil a \Rightarrow b$, то досить виразити через імплікацію заперечення. Тоді (з урахуванням результату прикладу 5) дана система функцій є повною. Проте заперечення не можна виразити через імплікацію. Справді, побудуємо таблицю:

a	$a \Rightarrow a$	$a \Rightarrow (a \Rightarrow a)$	$(a \Rightarrow a) \Rightarrow a$	$(a \Rightarrow a) \Rightarrow (a \Rightarrow a)$
0	1	1	0	1
1	1	1	1	1

Ми бачимо, що застосувавши імплікацію чи суперпозицію імплікацій, ми можемо одержати тільки тотожно істинну функцію або функцію, тотожно рівну a. Збільшення числа суперпозицій нічого не дасть, бо завжди можна замінити $a \Rightarrow (a \Rightarrow a) \equiv a \Rightarrow a$, $(a \Rightarrow a) \Rightarrow (a \Rightarrow a) \equiv a \Rightarrow a$, $(a \Rightarrow a) \Rightarrow a \equiv a$ і формулу, що містить суперпозицію довільного числа n імплікацій звести до формули, що містить не більше двох імплікацій. Отже, система $\{\Rightarrow\}$ неповна.

Вправи

1. Виразити:

a) $\wedge$ та $\Rightarrow$ через $\vee$ та $\rceil$;

b) $\vee$ та $\Rightarrow$ через $\wedge$та $\rceil$;

c) $\wedge$ та $\vee$ через $\Rightarrow$ та $\rceil$;

d) $\rceil$ через $\Rightarrow$ та 0;

e) $\vee$ через $\Rightarrow$.

2. Довести повноту систем функцій:

a) $\{\wedge, \vee, \rceil\}$; d) $\{\Rightarrow, \rceil\}$;
b) $\{\vee, \rceil\}$; e) $\{\Rightarrow, 0\}$;
c) $\{\wedge, \rceil\}$; f) $\{\downarrow\}$ (тут $a{\downarrow}b = \rceil(a \vee b)$).

3. Довести неповноту систем функцій:

a) $\{\wedge, \vee, \Rightarrow\}$; b) $\{\rceil\}$.

1.6. ЛОГІЧНЕ СЛІДУВАННЯ НА БАЗІ АЛГЕБРИ ВИСЛОВЛЕНЬ

Нехай A і B — формули алгебри висловлень, з яких хоча б одна містить пропозиційні змінні $p_1, ..., p_n$.

ОЗНАЧЕННЯ 1.6.1. *Кажуть, що формула B слідує логічно з формули A (на базі алгебри висловлень), якщо B набуває значення 1 при кожному розподілі істинісних значень $p_1, ..., p_n$, при якому A має значення 1. При цьому A називають посилкою, припущенням, гіпотезою, а B — логічним висновком.*

ОЗНАЧЕННЯ 1.6.2. *Кажуть, що формула B слідує логічно з посилок $A_1, ..., A_m$, якщо B набуває значення 1 при кожному такому розподілі істинісних значень $p_1, ..., p_n$, при якому всі посилки мають значення 1.*

Той факт, що B слідує з посилок $A_1, ..., A_m$, позначають $A_1, ..., A_m \models B$.

ТЕОРЕМА 1.6.1. *Формула B слідує логічно з посилок $A_1(p_1, ..., p_n), ..., A_m(p_1, ..., p_n\}$ тоді і тільки тоді, коли формула $A_1(p_1, ..., p_n) \wedge ... \wedge A_m(p_1, ..., p_n) \Rightarrow B$ є тавтологією.*

Достатність. Розглянемо довільний розподіл істинісних значень $p_1, ... p_n$, при якому всі A_i ($i = 1, ..., m$) набувають значення 1. Тоді $|A_1 \wedge ... \wedge A_m| = 1$ на розглядуваному наборі значень $p_1, ..., p_n$. Значення B на цьому наборі не може бути нулем, бо, за означенням імплікації, ми мали б $|A_1 \wedge ... \wedge A_m \Rightarrow B| = 0$ на розглядуваному наборі, що суперечить умові. Отже, на довільному наборі, де всі посилки A_i ($i=1, ..., m$) набувають значення 1, $|B| = 1$. Це означає, що B логічно слідує з посилок $A_1, ..., A_m$.

Необхідність. Дано

$$A_1, ..., A_m \models B. \tag{1}$$

Довести

$$\models A_1 \wedge ... \wedge A_m \Rightarrow B. \tag{2}$$

Припустимо, що (2) не є тавтологією. Тоді знайдеться хоча б один набір значень $p_1, ..., p_n$, на якому

$$|A_1 \wedge ... \wedge A_m \Rightarrow B| = 0. \tag{3}$$

На цьому наборі, за означенням імплікації, з (3) маємо

$$|A_i \wedge ... \wedge A_m| = 1, \quad (4)$$

$$|B| = 0. \quad (5)$$

З рівності (4) випливає, що всі A_i $(i = 1, ..., m)$ на цьому наборі мають значення 1. Враховуючи це і рівність (5), дістаємо, що B не слідує логічно з $A_1, ..., A_m$. Отже, ми зайшли у суперечність з (1). Наше припущення хибне, тобто співвідношення (2) доведено.

ТЕОРЕМА 1.6.2. *Відношення логічного слідування між формулами алгебри висловлень має наступі властивості:*

1) $A_1, A_2, ..., A_n \models A_m$ *для* $i = 1, 2, ..., n$;
2) *якщо* $A_1, ..., A_m \models B_j$ *для* $j=1,..., k$ *і* $B_1, ..., B_k$, *то* $A_1,..., A_m \models C$;
3) *якщо* $A_1, A_2, ..., A_m \models B$ *і* A *— довільна формула алгебри висловлень, то* $A_1, A_2, ..., A_m, A \models B$.

Доведення. 1. Фактично ця властивість полягає в наступному: $A_i \models A_i$. Вона безпосередньо випливає із означення 1.6.1 логічного слідування і означає, що відношення логічного слідування рефлексивне.

2. У частковому випадку при $m = k = 1$ дана властивість стверджує: якщо $A \models B$ і $B \models C$, то $A \models C$. Іншими словами, відношення логічного слідування транзитивне. Доведемо вихідне твердження. Будуємо таблицю істині для всіх формул, вказаних у пункті 2, перелічивши всі пропозиційні змінні хоча б в одній з цих формул, розглянемо який-небудь рядок цієї таблиці, в якій кожна формула $A_1,..., A_m$ набуває істинністне значення, рівне 1. Тоді на підставі умов кожна з формул $B_1, ..., B_k$ також приймає істинністне значення, рівне 1. Таким чином, для кожного набору істинностних значень змінних $p_1, p_2, ..., p_n$, на якому кожна формула $A_1, ..., A_m$ приймає значення 1, формула C також приймає значення 1. Це означає, що $A_1,.... A_m \models C$.

3. Пропонуємо читачеві самостійно довести цю властивість.

Наведемо тепер декілька схем логічного слідування.

1. $A \Rightarrow B, A \models B$ — правило висновку (modus ponens) або MP. Доводиться за допомогою тавтології 13 і теореми слідування.

2. $A \Rightarrow B, \urcorner B \models \urcorner A$ — modus tollens або MT. Доводиться за допомогою тавтології 14 і теореми слідування.

3. $A, B \models A \wedge B$ — правило введення кон'юнкції. Доводиться за допомогою тавтології 19 і теореми слідування.

4. $A \wedge B \models A, A \wedge B \models B$ — правило вилучення кон'юнкції. Доводиться за допомогою тавтології 6 і теореми слідування.

5. $A \models A \vee B, B \models A \vee B$ — правило виведення диз'юнкції. Доводиться за допомогою тавтології 7 і теореми слідування.

6. $A \Rightarrow B \models \urcorner B \Rightarrow \urcorner A$ — правило контрапозиції. Доводиться за допомогою тавтології 8 і теореми слідування.

7. $A \Rightarrow B, B \Rightarrow C \models A \Rightarrow C$ — правило силогізму. Доводиться за допомогою тавтології 9 і теореми слідування.

Приклад 1.6.1. Перевіримо, чи має місце логічне слідування $a \Rightarrow (\rceil b \vee c),\ \rceil a,\ b \Rightarrow c \models a \vee c$.

Припустимо, що $a \vee c$ не слідує логічно з попередніх формул. Отже, знайдуться конкретні висловлення a_0, b_0, c_0, такі, що $|a_0 \Rightarrow \rceil b_0 \vee c_0| = 1$, $|\rceil a_0| = 1$, $|b_0 \Rightarrow c_0| = 1$, але $|a_0 \vee c_0| = 0$. Тоді з останнього співвідношення отримаємо що $|a_0| = 0$ і $|c_0| = 0$. Далі із співвідношення $|b_0 \Rightarrow c_0| = 1$ отримаємо, що $|b| = 0$. Нарешті, знайшовши при даних значеннях $a_0,\ b_0,\ c_0$ значення $|a_0 \Rightarrow \rceil b_0|$ переконуємося: воно дорівнює 1, що знаходиться у повній відповідності з припущенням.

Отже, приходимо до висновку: якщо висловлення a_0, b_0, c_0, такі, що $|a_0| = |b_0| = |c_0| = 1$, то формули посилки приймуть значення 1, а формула $a \vee c$ має значення 0. Тому формула $a \vee c$ не слідує логічно з формул $a \Rightarrow (\rceil b \vee c),\ \rceil a,\ b \Rightarrow c$.

Вправи

1. Довести, що мають місце такі вивідності:

a) $a \models a$;

b) $a \Rightarrow b \models \rceil b \Rightarrow \rceil a$;

c) $a \Leftrightarrow b \models a \Rightarrow b$;

d) $(a \Leftrightarrow b) \wedge (b \Leftrightarrow c) \models a \Leftrightarrow c$;

e) $(a \Rightarrow b) \wedge (b \Rightarrow c) \models a \Rightarrow c$;

f) $(a \Rightarrow b) \models (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$;

g) $(a \Rightarrow b) \Rightarrow c) \models (a \Rightarrow b) \Rightarrow (a \Rightarrow c)$;

h) $(a \Rightarrow b) \Rightarrow c) \models b \Rightarrow (a \Rightarrow c)$.

2. Вкажіть правильні вивідності:

a) $a, b \models a \wedge b$;

b) $a \wedge b \models a$;

c) $a \wedge b \models b$;

d) $a \Rightarrow b, b \models a$;

e) $a \Rightarrow b,\ \rceil a \models \rceil b$;

f) $a \Rightarrow b \models \rceil a \Rightarrow \rceil b$;

g) $a \Leftrightarrow b, b \models a$;

h) $a \Leftrightarrow b \models \rceil b \Rightarrow \rceil a$;

i) $a \Leftrightarrow b \models \rceil a \Rightarrow \rceil b$;

j) $a \Leftrightarrow b \models \rceil a \Leftrightarrow \rceil b$.

3. Користуючись лише означенням логічного висновку на базі алгебри висловлень, для формул заданих таблицями:

a	b	c	α_1	α_2	α_3	β_1	β_2
0	0	0	1	0	1	1	0
0	0	1	0	1	1	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0
1	1	0	1	0	1	1	0
1	1	1	0	1	0	0	0

визначити, чи мають місце вивідності:

1) $\alpha_1, \alpha_3 \models \beta_1$, 2) $\alpha_1, \alpha_2, \alpha_3 \models \beta_1$, 3) $\alpha_1, \alpha_2 \models \beta_1$, 1) $\alpha_1, \alpha_2 \models \beta_2$?

Розділ 2

ЧИСЛЕННЯ ВИСЛОВЛЕНЬ

Опис алгебри висловлень та всі міркування про цю систему цілком задовольняють вимогам строгості, викладеним у вступі, оскільки вони ні в якій мірі не використовують поняття актуальної нескінченності, тобто є конструктивними.

Незважаючи на це, не завжди можна безпосередньо застосувати алгебру висловлень до висловлень математики. Щоб це можна було зробити завжди, ми повинні припустити, що кожне висловлення математики є або істинним, або хибним. Але подібне припущення є законом виключеного третього, який поширюється при цьому на нескінченну множину всіх висловлень. У такому вигляді цей принцип не може бути прийнятий в тій частині математики, яка ставить собі, зокрема, задачу обґрунтувати цей закон, показавши, що користування ним не приводить до суперечності.

Звідси виникає необхідність побудови аксіоматичної теорії (системи), яка відповідатиме алгебрі висловлень, будучи, разом з тим, вільною від вищезгаданої абстракції.

Цю формальну аксіоматичну систему будемо називати *числення висловлень*.

2.1. СИНТАКСИС ЧИСЛЕННЯ ВИСЛОВЛЕНЬ

Синтаксис будь-якого числення містить в собі опис символів цього числення, формул, вивідних формул.

Символи числення висловлень складаються із знаків трьох категорій (алфавіт числення висловлень):

1. Малі латинські літери з індексами та без них $a, b, c, \ldots, x, y, z, a_i, \ldots, z_1$, які називаються *пропозиційними літерами*.
2. Пропозиційні зв'язки: $\rceil, \wedge, \vee, \Rightarrow$.
3. (— ліва,) — права дужки.

Інших символів, крім вказаних, числення висловлень не має.

Введений алфавіт визначає штучну мову, яка називається *предметною мовою*. Мова, на якій описується предметна мова — *метамова*. У нашому випадку це українська мова.

Скінченні послідовності символів числення висловлень називаються *словами* числення висловлень. Не всяке слово, складене з вказаних символів, є формулою.

Повне означення формули має рекурсивний характер: вказують деякі вихідні формули та правила побудови, що дозволяють з формул утворювати нові формули.

Для позначення формул використовують літери $A, B, C, \ldots$ які не є символами числення висловлень.

ОЗНАЧЕННЯ 2.1.1. *1) Пропозиційна літера є формулою числення висловлень.*

2) Якщо A і B — формули, то слова $(\rceil A)$, $(\rceil B)$, $(A \wedge B)$, $(A \vee B)$, $(A \Rightarrow B)$, *— також формули числення висловлень.*

3) Інших формул числення висловлень, крім вказаних п. 1) і 2), немає.

Пункт 1) — базис, вказує вихідні формули, називаються *елементарними формулами,* п. 2) — індукційний крок, в ньому вказуються правила, які дозволяють з отриманих формул утворювати нові, п. 3) — обмеження.

Дужки дозволяють вказати порядок, в якому застосовувались правила побудови. Якщо при побудові формули A формула B використовується як елемент в індукційному правилі, то B називається *підформулою формули A*.

Множина формул є підмножиною множини всіх слів, побудованих з вихідних символів. Множина формул є *зчисленною*. Вона є також *рекурсивною*, тобто можна з'ясувати, чи є дане слово формулою, чи ні.

Введемо умови скорочення (опущення) дужок.

1. У запису формул зовнішні дужки не пишемо.
2. Ранг пропозиційних зв'язок $\rceil, \wedge, \vee, \Rightarrow$.

Виділимо деякий клас формул, які назвемо *вивідними* в численні висловлень.

Означення цих формул має рекурсивний характер. Спочатку визначимо вихідні вивідні формули — *аксіоми*, а потім сформулюємо правила, які дозволяють з даних формул утворювати нові — *правила виведення*.

Утворення вивідної формули з вихідних вивідних формул або аксіом, шляхом застосування правил виведення будемо називати *виведенням формули з аксіом*.

Аксіоми числення висловлень

I

А1. $a \Rightarrow (b \Rightarrow a)$;

А2. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$;

II

А3. $a \wedge b \Rightarrow a$;

А4. $a \wedge b \Rightarrow b$;

А5. $(a \Rightarrow b) \Rightarrow ((a \Rightarrow c) \Rightarrow (a \Rightarrow b \wedge c))$;

III

А6. $a \Rightarrow a \vee b$;

А7. $b \Rightarrow a \vee b$;

А8. $(a \Rightarrow c) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \vee b \Rightarrow c))$;

IV

А9. $(a \Rightarrow b) \Rightarrow (\urcorner b \Rightarrow \urcorner a)$;

А10. $a \Rightarrow \urcorner\urcorner a$;

А11. $\urcorner\urcorner a \Rightarrow a$.

Як видно із наведеного списку, аксіоми природньо розбиваються на чотири *групи*. Всі аксіоми групи I із логічних зв'язок містять тільки імплікацію. Ця зв'язка входить також в аксіоми інших груп. В групі II до імплікації додається кон'юнкція, в групі III — диз'юнкція, а в групі IV — заперечення.

Правила виведення

Операція підстановки це є відповідність, в якій кожній формулі A при заданій літері a та заданій формулі B відповідає цілком певна формула, яку ми позначимо $S_a^B(A)$.

Формулу $S_a^B(A)$ ми означимо таким чином:

якщо A є елементарна формула a , то $S_a^B(A)$ є формула B,
якщо A є відмінна від a пропозиційна літера q, то $S_a^B(A)$ є q;
$S_a^B(A \wedge C)$ є формула $S_a^B(A) \wedge S_a^B(C)$;
$S_a^B(A \vee C)$ є формула $S_a^B(A) \vee S_a^B(C)$;
$S_a^B(A \Rightarrow C)$ є формула $S_a^B(A) \Rightarrow S_a^B(C)$;
$S_a^B(\urcorner A)$ є формула $\urcorner S_a^B(A)$.

Якщо $A(a)$ — вивідна формула, то замінивши в ній a всюди, де вона входить, довільною формулою B, ми також отримаємо вивідну формулу.

Правило підстановки: *Якщо — A вивідна формула, то $S_a^B(A)$ теж вивідна, які б не були пропозиційна літера a та формула B.*

Правило підстановки скорочено будемо позначати ПП.

Приклад 2.1.1. Виконати підстановку: в аксіому *A1* замість літери a підставити формулу $a \wedge b \Rightarrow a$. (В подальшому таке завдання скорочено записуватимемо у вигляді $S_a^{a \wedge b \Rightarrow a}(A1)$).

Розв'язання. Виконавши вказану підстановку, одержимо $(a \wedge b \Rightarrow a) \Rightarrow (b \Rightarrow (a \wedge b \Rightarrow a))$.

Приклад 2.1.2. Яку підстановку і в яку аксіому було виконано, якщо в результаті одержано формулу $(a \wedge c \Rightarrow d) \Rightarrow (b \Rightarrow (a \wedge c \Rightarrow d))$?

Розв'язання. Помічаємо, що початок і кінець даної формули є однаковими підформулами, тобто дана формула має структуру $A \Rightarrow (b \Rightarrow A)$. Легко бачити, що таку структуру має аксіома 1. Отже, дана формула одержана з аксіоми 1 в результаті підстановки замість a формули A, тобто формули $a \wedge c \Rightarrow d$. Отже, відповідь $S_a^{a \wedge c \Rightarrow d}(A1)$.

Зауважимо, що правило підстановки дозволяє підставляти замість однієї літери (не замість формули) літеру. Якщо потрібно підставити замість двох (і більше) різних літер різні формули, то, згідно з правилом, необхідно виконувати підстановки поступово. У подальшому ланцюжки перетворень будемо записувати в стовпчик, так, щоб справа від кожної ланки дати її обґрунтування згідно означення.

Приклад 2.1.3. Підставити в *A7* замість a формулу $\neg a$, замість b формулу $b \vee c$.

Розв'язання.

1. $b \Rightarrow a \vee b$	A7;
2. $b \Rightarrow \neg a \vee b$	$S_a^{\neg a}(1)$;
3. $(a \vee b) \Rightarrow \neg a \vee (b \vee c)$	$S_b^{b \vee c}(2)$.

У побудованому стовпчику справа проведено аналіз: вказано, що формула є аксіомою (перший рядок) або ж показано, як вона одержується з попередніх формул. Запис $S_a^{\neg a}(1)$, як легко зрозуміти, означає: у формулі, що записана в першому рядку, замість a підставлено формулу $\neg a$.

Якщо в даній вправі виконати підстановки в іншому порядку (спочатку $S_b^{b \vee c}$, потім $S_a^{\neg a}$), то в цьому прикладі, як легко переконатись, одержимо ту ж саму формулу. Але змінюючи порядок підстановок формул, які містять ті ж літери, замість яких здійснюється підстановка, одержимо різні формули.

Приклад 2.1.4. Виконати підстановку $S_a^{a\Rightarrow b}(S_b^{a\wedge b}(A9))$ і порівняти одержаний результат з $S_b^{a\wedge b}(S_a^{a\Rightarrow b}(A9))$.

Розв'язання.

1. $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$ — $A9$;
2. $(a \Rightarrow a \wedge b) \Rightarrow (\neg(a \wedge b) \Rightarrow \neg a)$ — $S_b^{a\wedge b}(A9)$;
3. $((a\Rightarrow b)\Rightarrow(a\Rightarrow b)\wedge b)\Rightarrow(\neg((a\Rightarrow b)\wedge b)\Rightarrow\neg(a \Rightarrow b))$ — $S_a^{a\Rightarrow b}(S_b^{a\wedge b}(A9))$;

1°. $(a \Rightarrow b)\Rightarrow (\neg b \Rightarrow \neg a)$ — $A9$;
2°. $((a\Rightarrow b)\Rightarrow b)\Rightarrow (\neg a \Rightarrow \neg(a\Rightarrow b))$ — $S_a^{a\Rightarrow b}(A9)$;
3°. $((a\Rightarrow a\wedge b)\Rightarrow a\wedge b)\Rightarrow(\neg(a\wedge b)\Rightarrow\neg(a\Rightarrow a\wedge b))$ — $S_b^{a\wedge b}(S_a^{a\Rightarrow b}(A9))$.

Легко бачити, що формули 3 і 3° різні.

Зробимо послідовно підстановки у формулу A замість пропозиційних змінних $a_1, ..., a_n$. При цьому спочатку a_1 замінимо формулою B_1, потім в отриманій формулі a_2 замінимо формулою B_2 і т.д. В результаті отримаємо

$$S_{a_n}^{B_n}(S_{a_{n-1}}^{B_{n-1}}(...(S_{a_1}^{B_1}(A))...).$$

Якщо формули B_i не містять змінних a_1 ... a_n, то порядок виконання підстановок байдужий. В іншому випадку це не так. Проведемо таку операцію. Змінні a_i, у формулі A $(a_1,...,a_n)$ замінимо такими пропозиційними змінними, які не входять в жодну формулу B_i $(i = 1,..., n)$. Нехай це будуть пропозиційні змінні $x_1,..., x_n$. Тоді отримаємо формулу $A(x_1, ..., x_n)$. Після цього зробимо послідовні підстановки $S_{x_n}^{B_n}(S_{x_{n-1}}^{B_{n-1}}(...(S_{x_1}^{B_1}(A(x_1,...,x_n))..)$. Отримаємо формулу, яка є результатом одночасної заміни пропозиційних змінних $a_1, ..., a_n$ відповідно формулами $B_1, ..., B_n$. Отриману операцію позначимо:

$$S_{a_1...a_n}^{B_1...B_n}(A). \qquad (*)$$

Формула $A(x_1, ..., x_n)$ називається *вказівною формулою*. Операцію (*) будемо називати *складною підстановкою у формулу A* або просто *підстановкою у формулу A*.

Приклад 2.1.5. Виконати підстановку $S_{a,b}^{a\wedge b,a\vee b}(A8)$.

1. $(a \Rightarrow c) \Rightarrow ((b \Rightarrow c)\Rightarrow (a \vee b \Rightarrow c))$ — $A8$;
2. $(p \Rightarrow c) \Rightarrow ((q \Rightarrow c) \Rightarrow (p \vee q \Rightarrow c))$ — $S_{a,b}^{p,q}(1)$;
3. $(a \wedge b \Rightarrow c) \Rightarrow ((a \wedge b) \vee (a\vee b) \Rightarrow c))$ — $S_{p,q}^{a\wedge b,a\vee b}(A8)$.

Приклад 2.1.6. Яку підстановку і в яку аксіому було виконано, якщо в результаті одержано формулу:

а) $(a \wedge b \Rightarrow c)\Rightarrow (a \wedge b \Rightarrow c)\vee a \wedge c$?
б) $(a \wedge b \Rightarrow b)\Rightarrow ((c \Rightarrow b)\Rightarrow (a \wedge b \vee c \Rightarrow b))$?

Розв'язання. У випадку а) помічаємо, що формула ($a \wedge b \Rightarrow c$) повторюється, отже, ця формула має вигляд $A \Rightarrow A \vee a \wedge c$. Якщо позначити ($a \wedge c$) через B, маємо $A \Rightarrow A \vee B$. Таку структуру має аксіома А6. Отже, відповідь $S_{a,b}^{\,a\wedge b\Rightarrow c,a\wedge c}$ (А6).

У випадку б) важче помітити повторення підформули. Спробуємо встановити структуру формули трохи інакше. Випишемо дужки: $()\Rightarrow (()\Rightarrow ())$, причому кожна з формул, що стоїть в дужках, є імплікацією (остання операція в ній — імплікація). Схожу структуру мають 3 аксіоми: друга, п'ята, восьма. Аксіома А2 відпадає: у ній перша підформула ($a \Rightarrow (b \Rightarrow c)$) містить дві імплікації, в той час, як перша підформула даної формули ($a \wedge b \Rightarrow b$) містить тільки одну імплікацію. Аксіома А5 теж відпадає: у ній остання підформула ($a \Rightarrow b \wedge c$) має явно іншу структуру, ніж остання підформула даної формули ($a \wedge b \vee c \Rightarrow b$) (в першому випадку імплікується кон'юнкція, в другому пропозиційна змінна). Залишається аксіома 8. Порівнюємо підформули: $a \Rightarrow c$ та $a\wedge b \Rightarrow b$. Очевидно, $S_{a,c}^{a\wedge b,b}$; далі $b \Rightarrow c$ і $c \Rightarrow b$, що дає $S_{b,c}^{\,c,b}$; і нарешті $a\vee b \Rightarrow c$ і $a\wedge b\vee c\Rightarrow b$, що дає $S_{a,b,c}^{a\wedge b,c,b}$. Легко бачити, що в усіх трьох підформулах елементарні (прості) підстановки повторюються. Отже, відповідь: $S_{a,b,c}^{a\wedge b,c,b}$ ($A8$).

Основне правило виведення числення висловлень. *Якщо A та $A \Rightarrow B$ — вивідні формули числення висловлень, то B також вивідна формула. B — безпосередній висновок* (*заключення*). Називають modus ponens (MP). Правило MP часто називають *правилом відокремлення* або *правилом відривання*.

Приклад 2.1.7. Дано пари формул. До якої з них можна застосувати правило modus ponens? Яку формулу одержимо в результаті?

а) $A_1 = a \wedge b \Rightarrow c$; $A_2 = ((a \wedge b \Rightarrow c)\Rightarrow b \vee c)\Rightarrow ((\neg b \wedge \neg c)\Rightarrow a)$;
б) $A_1 = a \wedge b \Rightarrow c$; $A_2 = (a \wedge b \Rightarrow c)\wedge (\neg b \Rightarrow c \vee d)$;
в) $A_1 = a \wedge b \Rightarrow c$; $A_2 = (a \wedge b \Rightarrow c)\Rightarrow (b \vee c \Rightarrow ((\neg b \wedge \neg c)\Rightarrow a))$.

Розв'язання. У випадку а) правило modus ponens незастосовне: формула A_2 має не вигляд $A_1 \Rightarrow B$, а вигляд $(A_1 \Rightarrow B_1) \Rightarrow B$. У випадку б) правило modus ponens також незастосовне: формула має не вигляд $A \Rightarrow B$, а вигляд $A\wedge B$. Правило modus ponens застосовне лише у випадку в). В результаті його застосування одержимо формулу $b \vee c \Rightarrow ((\neg b \wedge \neg c)\Rightarrow a)$. Це будемо звичайно оформляти у вигляді:

1. $a \wedge b \Rightarrow c$	A_1;
2. $(a \wedge b \Rightarrow c)\Rightarrow (b \vee c \Rightarrow ((\neg b \wedge \neg c)\Rightarrow a))$	A_2;
3. $b \vee c \Rightarrow ((\neg b \wedge \neg c)\Rightarrow a)$	MP(1,2).

Тут запис MP(1,2) вказує, що формулу одержано за правилом заключення (modus ponens) із формул, що стоять в першому і другому рядках.

Завданням аксіом і правил виведення повністю визначаємо поняття вивідної в численні висловлень формули. Користуючись правилами виведення, можна, виходячи з аксіом, конструювати нові вивідні формули і отримати таким чином кожну вивідну формулу.

Приклад 2.1.8. Покажемо, що формула $a \wedge b \Rightarrow b \wedge a$ є вивідною.

Розв'язання. Вивчаємо структуру формули. Головна операція – імплікація. Цікавимося висновком імплікації. Це є кон'юнкція. Яка аксіома тут може знадобитись? Єдина аксіома, висновком імплікації в якої є кон'юнкція, це А5. Але посилкою для такого висновку в даній аксіомі є пропозиційна літера a, а нам потрібно $a \wedge b$. Отже зробимо підстановку $S_a^{a \wedge b}(A5)$, одержуємо формулу

$$(a \wedge b \Rightarrow b) \Rightarrow ((a \wedge b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow b \wedge c).$$

У цій формулі посилкою є А4. Отже, можемо застосувати правило MP, внаслідок чого отримаємо формулу $(a \wedge b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow b \wedge c)$. У цій формулі заключення є формула, яка відрізняється від даної тільки однією літерою, отже, зробимо підстановку замість пропозиційної літери c підставимо елементарну формулу a. Одержимо

$$(a \wedge b \Rightarrow a) \Rightarrow (a \wedge b \Rightarrow b \wedge a).$$

З цієї формули і А3 за правилом MP одержимо шукану формулу $a \wedge b \Rightarrow b \wedge a$.

Доведення оформляємо у такому вигляді:

1.	$(a \Rightarrow b) \Rightarrow ((a \Rightarrow c) \Rightarrow (a \Rightarrow b \wedge c)$	*A5;*
2.	$(a \wedge b \Rightarrow b) \Rightarrow ((a \wedge b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow b \wedge c))$	$S_a^{a \wedge b}(1)$
3.	$a \wedge b \Rightarrow b$	*A4;*
4.	$(a \wedge b \Rightarrow c) \Rightarrow (a \wedge b \Rightarrow b \wedge c)$	*MP(3,2);*
5.	$(a \wedge b \Rightarrow a) \Rightarrow (a \wedge b \Rightarrow b \wedge a)$	$S_c^a(4)$;
6.	$a \wedge b \Rightarrow a$	*A3;*
7.	$a \wedge b \Rightarrow b \wedge a$	*MP(6,5).*

Слід звернути увагу, що вказуючи до яких формул застосовується правило відокремлення (MP), першим вказуємо номер коротшої формули A, другим – номер довшої формули $A \Rightarrow B$. Або, іншими словами, перший номер – номер формули, яку "відриваємо", другий – номер формули, від якої "відриваємо".

ОЗНАЧЕННЯ 2.1.2. *Формальним доведенням формули числення висловлень B назвемо скінчену послідовність $A_1, \dots, A_n$ n формул, таку, що A_n співпадає з B, а кожний член послідовності є або аксіомою, або виводиться з попередніх членів за правилами виведення.*

ОЗНАЧЕННЯ 2.1.3. *Пропозиційна формула, для якої існує формальне доведення, називається теоремою числення висловлень.*

При доведенні формул (побудові виведення) звичайно доводиться використовувати обидва правила виведення, причому неодноразово. Першим, в основному, використовується правило підстановки. Спочатку з'ясовуємо, яку структуру має формула, котру маємо довести, і яка аксіома має схожу структуру. Шукаємо таку підстановку, щоб після її виконання одержати формулу, кінець якої співпадає з кінцем даної формули. Виконуємо цю підстановку і порівнюємо одержану формулу з потрібною. Далі можуть бути різні шляхи, які ми спробуємо пояснити на прикладах.

Приклад 2.1.9. Довести теорему (побудувати виведення):

$$(a \Rightarrow \neg b) \Rightarrow ((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)).$$

Розв'язання. Вивчаємо структуру формули. Помічаємо, що це є імплікація. Цікавимося кінцем (висновком імплікації). Це є імплікація з кон'юнкції. Яка аксіома тут може знадобитись? Є дві підходящих: А3 та А4. Кінець даної формули має вигляд $A \wedge b \Rightarrow A$. Отже, беремо аксіому А3: $a \wedge b \Rightarrow a$. Робимо підстановку $S_a^{c \Rightarrow b}$ (А3), одержуємо $(c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)$. Потрібна нам формула відрізняється від одержаної тільки початком $(a \Rightarrow \neg b)$. Щоб одержати з формули $(c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)$ формулу $(a \Rightarrow \neg b) \Rightarrow ((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b))$, застосуємо прийом, який будемо називати "приклеюванням початку". Беремо аксіому А1: $a \Rightarrow (b \Rightarrow a)$. Замість a підставляємо формулу, яку ми вже одержали $(c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)$, а замість b підставляємо формулу, яку треба "приклеїти" $(a \Rightarrow \neg b)$, тобто $S_{a,b}^{(c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b),\, a \Rightarrow \neg b}$ (А1). Одержимо $((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)) \Rightarrow ((a \Rightarrow \neg b) \Rightarrow ((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)))$. З даної формули і формули $(c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)$ за правилом заключення (modus ponens) одержимо шукану формулу $(a \Rightarrow \neg b) \Rightarrow ((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b))$.

Доведення оформляємо у такому вигляді:

1. $a \wedge b \Rightarrow a$ — А3;
2. $(c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)$ — $S_a^{c \Rightarrow b}(1)$;
3. $a \Rightarrow (b \Rightarrow a)$ — А1;
4. $((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)) \Rightarrow ((a \Rightarrow \neg b) \Rightarrow ((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b)))$ — $S_{a,b}^{(c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b),\, a \Rightarrow \neg b}(3)$;
5. $(a \Rightarrow \neg b) \Rightarrow ((c \Rightarrow b) \wedge b \Rightarrow (c \Rightarrow b))$ — MP(2,4).

ТЕОРЕМА 2.1.1. $a \Rightarrow a$.

Доведення.

1. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$ A2;
2. $(a \Rightarrow (b \Rightarrow a)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow a))$ $S_c^a(1)$;
3. $a \Rightarrow (b \Rightarrow a)$ A1;
4. $(a \Rightarrow b) \Rightarrow (a \Rightarrow a)$ MP(3,2);
5. $(a \Rightarrow (b \Rightarrow a)) \Rightarrow (a \Rightarrow a)$ $S_b^{b \Rightarrow a}(4)$;
6. $a \Rightarrow a$ MP(3,5).

Отримали послідовність формул 1-6, яка є формальним доведенням формули $a \Rightarrow a$.

Розглянемо прийом, який часто застосовується для побудови формального доведення. У теоремі 2.3.1 цей прийом буде сформульований як правило силогізму. Коротко його суть така. Треба довести формулу $A \Rightarrow C$. А вдалось одержати формули $A \Rightarrow B$ та $B \Rightarrow C$. Оскільки є тільки імплікація, то слід використати аксіоми першої групи. $A1$ не підходить – вона містить 2 літери, а маємо три різні підформули A, B, та C. Отже беремо $A2$: $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$ і виконуємо підстановку. Оскільки нам треба одержати формулу $A \Rightarrow C$, а остання підформула $A2$ є $a \Rightarrow c$, то виконаємо $S_{a,c}^{A,C}(A2)$. Тоді замість b слід підставити B. Отже $S_{a,b,c}^{A,B,C}(A2)$

$$(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C)).$$

Шукана формула є останньою підформулою одержаної. Отже, щоб довести її, слід "відірвати" початок, як описано у прикладі 7.3. Отже, треба показати вивідність формул $A \Rightarrow (B \Rightarrow C)$ та $A \Rightarrow B$. Остання вивідна за умовою, що ж стосується формули $A \Rightarrow (B \Rightarrow C)$, то вона одержується з вивідної за умовою формули $B \Rightarrow C$ прийомом "істина з чого завгодно". Беремо A_1 і робимо підстановку $S_{a,b}^{B \Rightarrow C, A}(A1)$, отримаємо формулу $(B \Rightarrow C) \Rightarrow (A \Rightarrow (B \Rightarrow C)$. З останньої за правилом MP отримаємо формулу $A \Rightarrow (B \Rightarrow C)$.

Проілюструємо розказане на прикладі.

Приклад 2.1.10. Довести теорему $a \wedge b \Rightarrow a \vee c$.

Розв'язання. Помічаємо, що формули $a \wedge b \Rightarrow a$ та $a \Rightarrow a \vee c$ вивідні (перша формула – А3, друга отримана з А6 шляхом підстановки). Не вдаючись у детальні пояснення, які наведено вище у загальному вигляді, запишемо лише вивід з аналізом:

1. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$ A2;

2. $(a\wedge b\Rightarrow(a\Rightarrow a\vee c))\Rightarrow((a\wedge b\Rightarrow a)\Rightarrow(a\wedge b\Rightarrow a\vee c))$ $\quad S_{a,b,c}^{a\wedge b,a,a\vee c}(1)$;
3. $a\Rightarrow a\vee b$ $\quad$ A6;
4. $a\Rightarrow a\vee c$ $\quad S_b^c(3)$;
5. $a\Rightarrow(b\Rightarrow a)$ $\quad$ A1;
6. $(a\Rightarrow a\vee c)\Rightarrow(a\wedge b\Rightarrow(a\Rightarrow a\vee c))$ $\quad S_{a,b}^{a\Rightarrow a\vee b,a\wedge b}(5)$;
7. $a\wedge b\Rightarrow(a\Rightarrow a\vee c)$ $\quad$ MP(4,6);
8. $(a\wedge b\Rightarrow a)\Rightarrow(a\wedge b\Rightarrow a\vee c)$ $\quad$ MP(7,2);
9. $a\wedge b\Rightarrow a$ $\quad$ A3;
10. $a\wedge b\Rightarrow a\vee c$ $\quad$ MP(9,8).

Отже, основне при побудові виведення — з'ясувати структуру формули, вдало вибрати аксіому і підстановку.

Вправи

1. Довести теореми числення висловлень:

a) $a\vee b\Rightarrow b\vee a$;
b) $a\wedge b\Rightarrow b\vee a$;
c) $a\Rightarrow a\wedge a$;
d) $(a\wedge a\Rightarrow a)(a\Rightarrow a\wedge a)$;
e) $(a\Rightarrow b)\vee(b\Rightarrow a)$;
f) $\neg a\vee a$.

2. Довести теореми числення висловлень:

a) $(b\Rightarrow(a\Rightarrow(b\Rightarrow c)))\Rightarrow(b\Rightarrow((a\Rightarrow b)\Rightarrow(a\Rightarrow c)))$;
b) $(a\Rightarrow b)\Rightarrow(a\Rightarrow\neg\neg b)$;
c) $(a\Rightarrow\neg\neg b)\Rightarrow(a\Rightarrow b)$;
d) $a\vee c\Rightarrow((a\vee b\Rightarrow c)\Rightarrow(\neg c\Rightarrow\neg(a\vee b)))$;
e) $(b\Rightarrow a\vee c)\Rightarrow((b\Rightarrow c)\Rightarrow((d\Rightarrow c)\Rightarrow(b\vee d\Rightarrow c)))$;
f) $(\neg a\Rightarrow b)\Rightarrow(\neg b\Rightarrow(\neg\neg a\Rightarrow a))$.

2.2. ВИВІДНІСТЬ З СИСТЕМ ФОРМУЛ

В математиці зустрічаються твердження про те, що об'єкти мають властивості при виконанні певних умов. Наприклад "Якщо дуги кіл однакового радіусу рівні, то і хорди, які їх стягують, рівні". Тут твердження "Якщо дуги рівні, то і хорди, які їх стягують, рівні" справедливе лише при виконанні певної умови: взято дуги (і хорди) одного кола або рівних кіл. Інший приклад "В рівнобедреному трикутнику бісектриса кута при вершині є також медіаною, проведеною до основи". Тут твердження "Якщо відрізок є бісектрисою кута трикутника, то він є також медіаною, проведеною до протилежної сторони", справджується при виконанні двох умов: трикутник рівнобедрений (дві сторони рівні), кут взято проти третьої сторони.

У формальній логіці ця ситуація описується вивідністю з систем формул. Замість того, щоб вивідні в численні висловлень формули виводити з аксіом, застосовуючи безпосередньо правила виведення, обираємо більш короткий шлях.

ОЗНАЧЕННЯ 2.2.1. *Формула B вивідна з формул $A_1, ... A_n$, які називають посилками, припущеннями, вихідними формулами, якщо існує скінчена послідовність формул*

$$B_1, ..., B_i, ..., B_m \tag{2.2.1}$$

така, що B_m збігається з B та кожен член послідовності (2.2.1) є або аксіомою, або одним з припущень, або безпосереднім висновком з попередніх членів за правилом MP.

Позначимо $\Gamma = \{A_1, ..., A_n\}$, вивідність B з Γ записується так:

$$A_1, ..., A_n \vdash B \text{ або } \Gamma \vdash B. \tag{2.2.2}$$

Мають місце такі твердження:

а) кожна формула A_i ($i = 1, ..., n$) вивідна з $A_1, ..., A_n$;

б) кожна формула, вивідна у численні висловлень, вивідна з $A_1, ..., A_n$;

в) якщо $\Gamma \vdash B$, то $\Gamma, A \vdash B$, де A — довільна формула числення висловлень.

Розширимо поняття вивідності (2.2.2) на випадок, коли $\Gamma = \varnothing$. Тоді маємо $\vdash B$ і вважаємо, що B є просто вивідною формулою (теоремою числення висловлень).

Приклад 2.2.1. З системи формул $a \vee b \Rightarrow (c \Rightarrow d)$, $(c \Rightarrow d) \Rightarrow (a \wedge \neg b \Rightarrow c)$, $a \vee b$ вивести формулу $a \wedge \neg b \Rightarrow c$. (Умову можна записати так: $a \vee b \Rightarrow (c \Rightarrow d), (c \Rightarrow d) \Rightarrow (a \wedge \neg b \Rightarrow c), a \vee b \vdash a \wedge \neg b \Rightarrow c$).

Розв'язання. Оскільки міркування, що приводять до виводу, дуже прості, ми їх не наводимо. Подаємо лише виведення з аналізом.

1. $a \vee b \Rightarrow (c \Rightarrow d)$	A_1 — дана формула;
2. $a \vee b$	A_3 — дана формула;
3. $c \Rightarrow d$	MP(2,1);
4. $(c \Rightarrow d) \Rightarrow (a \wedge \neg b \Rightarrow c)$	A_2 — дана формула;
5. $a \wedge \neg b \Rightarrow c$	MP(3,4).

Для зручності аналізу формули даної системи ми занумерували відповідно A_1, A_2, A_3.

В цьому прикладі ми вивели формулу $B = a \wedge \neg b \Rightarrow c$ з формул даної системи, не використовуючи інших формул (ні аксіом, ні теорем). Так буває не завжди.

Приклад 2.2.2. Довести вивідність:

$$a \Rightarrow b \vee d,\ b \vee d \Rightarrow ((a \Rightarrow \neg c) \Rightarrow d),\ a \wedge b \vdash (a \Rightarrow \neg c) \Rightarrow d.$$

Розв'язання. Щоб довести вивідність, досить побудувати виведення формули B. Серед даних формул є формула виду $A \Rightarrow B$. Отже, для вивідності B слід спочатку вивести $A = b \vee d$. Але остання формула є висновком імплікації і тепер задача зводиться до виведення формули a, але цієї елементарної формули серед формул даної системи немає. Її можна одержати, використовуючи дану формулу $a \wedge b$ і А3. Запишемо виведення:

1.	$a \Rightarrow b \vee d$	припущення;
2.	$b \vee d \Rightarrow ((a => \neg c) \Rightarrow d)$	припущення;
3.	$a \wedge b$	припущення;
4.	$\vdash a \wedge b \Rightarrow a$	А3;
5.	$\vdash a$	МР(3,4);
6.	$\vdash b \vee d$	МР(5,1);
7.	$\vdash (a \Rightarrow \neg c) \Rightarrow d$	МР(6,2).

Отже, за означенням вивідності з послідовності формул 1-7 (Df(l-7)), маємо:

$$a \Rightarrow b \vee d,\ b \vee d \Rightarrow ((a \Rightarrow \neg c),\ a \wedge b \vdash (a \Rightarrow \neg c) \Rightarrow d.$$

Зауважимо, що при побудові виведення з системи формул можна використовувати тільки правило відривання МР. Правило підстановки тут заборонено. Точніше, оскільки ми використовуємо при виведенні теореми і аксіоми, то ми можемо робити довільні підстановки в теореми і аксіоми. Але у вихідні формули (формули системи Γ) і у формули, одержані з вихідних за правилом МР, жодних підстановок робити не можна.

Отже, при побудові виведення з системи формул треба уважно стежити чи не одержана формула, в яку ми зробили підстановку, з формул системи Γ.

Приклад 2.2.3. Перевірити виведення з аналізом.

1. $a \Rightarrow a \vee b$	А6;
2. $b \Rightarrow b \vee c$	$S_{a,b}^{b,c}$ (1);
3. $b \Rightarrow a \vee b$	А7;
4. $c \Rightarrow b \vee c$	$S_{a,b}^{b,c}$ (3);
5. $(a \Rightarrow c) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \vee b \Rightarrow c))$	А8;
6. $(a \Rightarrow b) \Rightarrow ((c \Rightarrow b) \Rightarrow (a \vee c \Rightarrow b))$	$S_{b,c}^{c,b}$ (5);
7. $a \Rightarrow b$	A_1 – дана формула;
8. $(c \Rightarrow b) \Rightarrow (a \vee c \Rightarrow b)$	МР(7,6);
9. $(c \Rightarrow b \vee c) \Rightarrow (a \vee c \Rightarrow b \vee c)$	$S_b^{b \vee c}$ (8);
10. $a \vee c \Rightarrow b \vee c$	МР(4,9).

Розв'язання. Вправу виконано неправильно, оскільки на дев'ятому кроці виконувалась підстановка в формулу $(c \Rightarrow b) \Rightarrow (a \vee c \Rightarrow b)$, яка одержана за правилом відривання даної формули $a \Rightarrow b$ (і теореми). Для правильного розв'язання вправи слід було б:

6'. $(a\Rightarrow b\vee c)\Rightarrow((c\Rightarrow b\vee c)\Rightarrow(a\vee c\Rightarrow b\vee c))$	$S_{c,b}^{b\vee c,c}$ (5);
7'. $(a\Rightarrow (b\Rightarrow c))\Rightarrow ((a\Rightarrow b)\Rightarrow (a\Rightarrow c))$	A2;
8'. $(a\Rightarrow(b\Rightarrow b\vee c))\Rightarrow((a\Rightarrow b)\Rightarrow(a\Rightarrow b\vee c))$	$S_c^{b\vee c}$ (7');
9'. $a\Rightarrow (b\Rightarrow a)$	A1;
10'. $(b\Rightarrow b \vee c)\Rightarrow (a\Rightarrow (b\Rightarrow b \vee c))$	$S_{a,b}^{b\Rightarrow b\vee c,a}$ (9');
11'. $a\Rightarrow (b\Rightarrow b \vee c)$	MP(2,10');
12'. $(a\Rightarrow b)\Rightarrow (a\Rightarrow b \vee c)$	MP(11',8');
13'. $a\Rightarrow b$	A_1 — дано;
14'. $a\Rightarrow b \vee c$	MP(13',12');
15'. $(c\Rightarrow b \vee c)\Rightarrow (a \vee c\Rightarrow b \vee c)$	MP(14',6');
16'. $a \vee c\Rightarrow b \vee c$	MP(4,15').

Метатеорема дедукції не є теоремою числення висловлень. Це теорема про числення висловлень. З її допомогою можна довести, що для деякої формули існує формальне доведення (тобто, що формула є теоремою). Якщо ж ми хочемо побудувати це доведення, треба використовувати не саму метатеорему дедукції (не її результат), а метод, яким вона була доведена.

МЕТАТЕОРЕМА ДЕДУКЦІЇ (МТД). *Якщо Γ — множина формул числення висловлень, A і B — довільні його формули та $\Gamma, A \vdash B$, то $\Gamma \vdash A \Rightarrow B$.*

Доведення. За умовою, існує така послідовність формул:

$$B_1, ..., B_I , ..., B_m, \tag{2.2.3}$$

де $B_m = B$, кожна B_i ($i = 1, ..., m$) є або 1) аксіомою, або 2) одним з припущень, або 3) безпосереднім висновком з попередніх членів, тобто існують j, $k < i$ такі, що $B_k = B_j \Rightarrow B_i$. Покажемо, що у кожному з цих трьох єдино можливих випадків має місце вивідність $\Gamma \vdash A \Rightarrow B_i$, для кожного i ($i = 1, ..., m$).

1). Якщо B_i, — аксіома, то поповнимо послідовність (2.2.3) такими формулами, вивідними з Γ

$m + 1$ B_i	аксіома;
$m + 2$ $B_i \Rightarrow (A \Rightarrow B_i)$	$S_{a,b}^{B_i,A}(A1)$;
$m + 3$ $A \Rightarrow B_i$	MP($m + 1$, $m + 2$).

2а). Якщо B_i співпадає з A, то $A \Rightarrow A$ є $S_a{}^A$ (Т7.1) — вивідна формула, тому $\Gamma \vdash A \Rightarrow B_i$.

2б). Нехай $B_i \in \Gamma$, тоді приєднаємо до формул (2.2.3) послідовно формули B_i (припущення — елемент з Γ), $B_i \Rightarrow (A \Rightarrow B_i,)$ — $S_{a,b}^{B_i,A}(A1)$ ця формула, як вивідна, виводиться з Γ, тому формула $A \Rightarrow B_i$, виводиться з Γ за правилом MP.

3). Доведення проведемо за індукцією. Індуктивне припущення, що для всіх натуральних $s<i$ має місце вивідність $\Gamma \vdash A \Rightarrow B_s$. Тоді $\Gamma \vdash A \Rightarrow B_j$, і $\Gamma \vdash A \Rightarrow B_k$ *бо* $j< i,\ k < i$. Далі:

1.	Γ	множина припущень;
2.	$(A\Rightarrow(B_j\Rightarrow B_i)\Rightarrow((A\Rightarrow B_j)\Rightarrow(A\Rightarrow B_i)$	$S_{a,b,c}^{A,B_j,B_i}(A2)$;
3.	$A \Rightarrow (B_j \Rightarrow B_i)$	індуктивне припущення: $B_k = B_j \Rightarrow B_i$;
4.	$(A \Rightarrow B_j) \Rightarrow (A \Rightarrow B_i)$	MP(3,2);
5.	$(A \Rightarrow B_j)$	індуктивне припущення;
6.	$(A \Rightarrow B_i)$	MP(5,4).

Отже, з індуктивного припущення випливає, що $\Gamma \vdash A \Rightarrow B_i$.

Тепер, за принципом індукції, твердження $\Gamma \vdash A \Rightarrow B_i$, у випадку 3) виконується для всіх натуральних чисел, зокрема, також для числа i, тобто $\Gamma \vdash A \Rightarrow B_i$,в усіх трьох випадках для довільного i $(i = 1, 2, ..., m)$. Як окремий випадок, при $i = m$, дістанемо $\Gamma \vdash A \Rightarrow B_m$, або $\Gamma \vdash A \Rightarrow B$.

Приклад 2.2.4. Відновлюючи хід доведення метатеореми дедукції, але не використовуючи самої теореми, довести, що з вивідності $a \Rightarrow b, a \vdash b \vee c$ випливає $a \Rightarrow b \vdash a\Rightarrow b \vee c$.

Розв'язання. Перш за все перевіримо істинність вивідності $a \Rightarrow b, a \vdash b \vee c$.

1. $a\Rightarrow b$	A_1	Γ_0;
2. a	A_2	Γ_0;
3. b	MP(2,1)	Γ_1;
4. $a \Rightarrow a \vee b$	A6	Γ_0;
5. $b \Rightarrow b \vee c$	$S_{a,b}^{b,c}$ (4)	Γ_0;
6. $b \vee c$	MP(3,5)	Γ_2.

Отже, формула $b \vee c$ дійсно виводиться з формул $a \Rightarrow b$, a, причому виводиться на другому кроці. Це означає, що в множині Γ, є формули b та $b \Rightarrow b \vee c$ (друга належить Γ_0 , але за домовленістю $\Gamma_0 \subset \Gamma_1$), безпосереднім висновком яких є формула $b \vee c$. Справедливі вивідності $a \Rightarrow b \vdash a \Rightarrow b$ (дана формула) та $a \Rightarrow b \vdash a \Rightarrow (b \Rightarrow b \vee c)$, оскільки формула $a \Rightarrow (b \Rightarrow b \vee c)$ є теоремою. Доведемо тепер вивідність $a \Rightarrow b \vdash a \Rightarrow b \vee c$, використовуючи хід доведення метатеореми дедукції. Тут $\Gamma = \{a \Rightarrow b\}$, $A = a$, $B = b \vee c$.

1. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$ A2;
2. $(a \Rightarrow (b \Rightarrow b \vee c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow b \vee c))$ $S_c^{b\vee c}(1)$.

(Тут формули $A = a$ та $B_1 = b$ не потребують підстановки).

3. $a \Rightarrow (b \Rightarrow b \vee c)$ теорема.

(При потребі можна дати її виведення. Воно включає A6, $S_{a,b}^{b,c}$(A6), A1, $S_{a,b}^{b\Rightarrow b\vee c,a}$(A1), дану формулу).

4. $(a \Rightarrow b) \Rightarrow (a \Rightarrow b \vee c)$ MP(3,2);
5. $a \Rightarrow b$ дана формула;
6. $a \Rightarrow b \vdash a \Rightarrow b \vee c$ MP(5,4).

Розглянемо тепер зразок доведення теореми числення висловлень за допомогою МТД. Зауважимо, що в цьому випадку ми не будуємо формальне доведення теореми, а лише стверджуємо, що таке доведення існує.

ТЕОРЕМА 2.2.1. $\vdash (a \Rightarrow (b \Rightarrow c)) \Rightarrow (a \wedge b \Rightarrow c)$. (2.2.4)

Доведення. Використання МТД полягає в тому, що ми виділяємо деяку систему формул, з якої треба вивести останню підформулу імплікації.

Покажемо, що з системи формул $a \Rightarrow (b \Rightarrow c)$, $a \wedge b$ виводиться формула c, тобто $a \Rightarrow (b \Rightarrow c), a \wedge b \vdash c$. Справді, формула c є безпосереднім наслідком формул $b \Rightarrow c$ та b (одержується з них за правилом MP). Формулу $b \Rightarrow c$ можна також одержати за правилом MP з першої формули даної системи і формули a. Останню формулу і елементарну формулу b можна одержати відповідно з A3 і A4 та другої формули даної системи. Доведення оформимо у вигляді:

1. $a \Rightarrow (b \Rightarrow c)$ припущення;
2. $a \wedge b$ припущення;
3. $\vdash a \wedge b \Rightarrow a$ A3;
4. $\vdash a \wedge b \Rightarrow b$ A4;
5. $\vdash a$ MP(2,3);
6. $\vdash b$ MP(2,4);
7. $\vdash b \Rightarrow c$ MP(5,1);
8. $\vdash c$ MP(6,7).

За означенням вивідності з 1-8 маємо:

9. $a \Rightarrow (b \Rightarrow c), a \wedge b \vdash c$ Df(1-8);

звідки, застосовуючи метатеорему дедукції, дістанемо

10. $a \Rightarrow (b \Rightarrow c), \vdash a \wedge b \Rightarrow c$ МТД(9);
11. $\vdash (a \Rightarrow (b \Rightarrow c)) \Rightarrow (a \wedge b \Rightarrow c)$ МТД(10).

Для виділення множини формул Γ і формули B звичайно діємо так: знаходимо головну операцію у формулі, для якої будуємо виведення. Якщо це імплікація, то посилку цієї імплікації (в Т2.2.1 це $a \Rightarrow (b \Rightarrow c))$ позначимо $A_1 \in \Gamma$. Розглядаємо висновок (в Т2.2.1 це $a \wedge b \Rightarrow c$). Якщо ми бачимо, як вивести цю формулу з формули A_1 і аксіом, то позначаємо висновок через B і будуємо доведення. Якщо ж ми не можемо довести вивідність відразу, і, якщо головна операція у висновку імплікація, то посилка її $a \wedge b$ дає $A_2 \in \Gamma$. Якщо ми вміємо довести вивідність висновку з побудованої системи формул A_1, A_2 то позначаємо цей другий висновок через B і доводимо вивідність.

Якщо ж ми не можемо довести вивідність відразу, то продовжуємо побудову системи *Г*, відділяючи від одержаного висновку нову формулу, яка є в ньому посилкою. Цей процес обов'язково закінчиться: на якомусь кроці одержимо формулу, в якій операція імплікації не є головною. Це і буде формула *В*.

Проілюструємо це на конкретному доведенні.

ТЕОРЕМА 2.2.2. $\vdash (a\Rightarrow (b\Rightarrow c))\Rightarrow (b\Rightarrow (a \Rightarrow c))$. (2.2.5)

Шукаємо головну операцію імплікації і посилку цієї імплікації $a \Rightarrow (b \Rightarrow c)$ позначимо A_1. Висновок $b \Rightarrow (a \Rightarrow c)$ має головну операцію імплікація. Посилку – b позначимо A_2. Висновок $a \Rightarrow c$ також має головною операцією імплікацію, посилку a позначимо A_3. Отже, задача доведення Т2.2.2 звелась до доведення вивідності $a \Rightarrow (b \Rightarrow c), b, a \vdash c$.

1. $a \Rightarrow (b \Rightarrow c)$	припущення;
2. b	припущення;
3. a	припущення;
4. $\vdash b \Rightarrow c$	MP(3,1);
5. $\vdash c$	MP(2,4);
6. $a \Rightarrow (b \Rightarrow c), b, a \vdash c$	Df(1-5);
7. $a \Rightarrow (b \Rightarrow c), b \vdash a \Rightarrow c$	МТД(6);
8. $a \Rightarrow (b \Rightarrow c) \vdash b \Rightarrow (a \Rightarrow c)$	МТД(7);
9. $\vdash (a \Rightarrow (b \Rightarrow c)) \Rightarrow (b \Rightarrow (a \Rightarrow c))$.	МТД(8).

МЕТАТЕОРЕМА (обернена до теореми дедукції – ОТД). *Якщо $Г \vdash A \Rightarrow B$, то $Г, A \vdash B$.*

Доведення.

1. $Г$	множина припущень;
...	
n+1. A	припущення;
n+2. $\vdash A \Rightarrow B$	за умовою теореми, з $(1-n)$;
n+3. $\vdash B$	$MP(n+1, n+2)$.

Отже, за означенням вивідності $(Df(l - n+3))$ маємо: $Г, A \vdash B$.

Вправи

1. Обґрунтувати вивідність:

a) $a\Rightarrow b, b\Rightarrow c \vdash a\Rightarrow c$;	f) $a\Rightarrow b \vdash (b\Rightarrow c)\Rightarrow(a\Rightarrow c)$;
b) $a\Rightarrow(b\Rightarrow c) \vdash b\Rightarrow(a\Rightarrow c)$;	g) $a\Rightarrow b \vdash (c\Rightarrow a)\Rightarrow(c\Rightarrow b)$;
c) $a\Rightarrow(b\Rightarrow c) \vdash a\wedge b\Rightarrow c$;	h) $a\Rightarrow b \vdash c\wedge a\Rightarrow c\wedge b$;
d) $a\Rightarrow(b\Rightarrow c) \vdash b\Rightarrow(a\Rightarrow c)$;	i) $a\Rightarrow b \vdash c\vee a\Rightarrow c\vee b$;
e) $a\wedge b\Rightarrow c \vdash a\Rightarrow(b\Rightarrow c)$;	j) $a\Rightarrow b \vdash a\vee c\Rightarrow b\vee c$.

2. Побудувати виведення:

a) $c \Rightarrow a,\ c \Rightarrow b,\ c \vdash a \wedge b;$
b) $a \wedge b,\ a \Rightarrow (b \Rightarrow c) \vdash c;$
c) $(a \Rightarrow c) \Rightarrow c,\ a \Rightarrow b,\ b \Rightarrow c \vdash c;$
d) $a \Rightarrow b,\ \overline{b} \vdash \overline{a};$
e) $a \Rightarrow b,\ c \Rightarrow d,\ \overline{b} \wedge \overline{d} \vdash \overline{a} \wedge \overline{c};$
f) $a \Rightarrow c,\ \overline{b} \Rightarrow a \vdash \overline{b} \Rightarrow c;$
g) $a \Rightarrow b,\ a \Rightarrow \overline{b} \vdash \overline{a} \vee b;$
h) $\overline{a} \Rightarrow b,\ b \Rightarrow c \vdash a \Rightarrow c;$
i) $b \Rightarrow \overline{a},\ a \vdash \overline{b};$
j) $a \Rightarrow b,\ a \Rightarrow \overline{b} \vdash \overline{a}.$

2.3. ПОХІДНІ ПРАВИЛА ЧИСЛЕННЯ ВИСЛОВЛЕНЬ

При доведенні теорем поряд з основними правилами виведення (підстановки і modus ponens) використовують також похідні правила виведення, які є метатеоремами числення висловлень. Використання похідних правил виведення полегшує і скорочує доведення теорем, проте при цьому ми не одержуємо виведення формули, а лише доводимо, що таке виведення існує. Щоб на основі цього доведення побудувати формальне виведення, треба кожне використання похідних правил замінити доведенням цього правила на даному конкретному матеріалі. Прикладом похідного правила виведення може служити теорема дедукції — правило введення імплікації. Похідним правилом виведення є також правило одночасної підстановки, яке скорочує доведення. Крім названих вище, звичайно використовують інші похідні правила, мета яких полягає в скороченні числа кроків формального доведення. Інакше кажучи, похідні правила це будівельні блоки при побудові формального доведення, схеми, кожна з яких замінює кілька кроків доведення.

ТЕОРЕМА 2.3.1. $\vdash (a \Rightarrow b) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \Rightarrow c)).$ (2.3.1)

1. $a \Rightarrow b$ припущення;
2. $b \Rightarrow c$ припущення;
3. a припущення;
4. $\vdash b$ MP(3,1);
5. $\vdash c$ MP(4,2);
6. $a \Rightarrow b,\ b \Rightarrow c,\ a \vdash c$ Df(1-5);
7. $a \Rightarrow b,\ b \Rightarrow c \vdash a \Rightarrow c$ МТД(6);
8. $a \Rightarrow b \vdash (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$ МТД (7);
9. $\vdash a \Rightarrow b) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \Rightarrow c))$ МТД (8).

Зробимо підстановку $S_{a,b,c}^{A,B,C}(A2)$ отримаємо:

$$\vdash (A \Rightarrow B) \Rightarrow ((B \Rightarrow C) \Rightarrow (A \Rightarrow C)). \qquad (2.3.2)$$

Отже, якщо $A \Rightarrow B,\ B \Rightarrow C$ вивідні, то з (2.3.2) із правилом MP, $A \Rightarrow C$ — вивідна. Звідси маємо *правило силогізму*.

$A \Rightarrow B,\ B \Rightarrow C \vdash A \Rightarrow B$ — правило силогізму (ПС).

ТЕОРЕМА 2.3.2. $\vdash a \Rightarrow (b \Rightarrow a \wedge b)$. (2.3.3)

1. a припущення;
2. b припущення;
3. $\vdash a \Rightarrow (b \Rightarrow a)$ A1;
4. $\vdash b \Rightarrow a$ MP (1,3);
5. $\vdash b \Rightarrow b$ T7.1;
6. $\vdash (b \Rightarrow a) \Rightarrow (b \Rightarrow b) \Rightarrow (b \Rightarrow a \wedge b))$ $S_{a,b,c}^{b,a,b}(A5)$;
7. $\vdash (b \Rightarrow b) \Rightarrow (b \Rightarrow a \wedge b)$ MP(4.6);
8. $\vdash b \Rightarrow a \wedge b$ MP(5,7);
9. $\vdash a \wedge b$ MP(2, 8);
10. $a, b \vdash a \wedge b$ Df(l-9);
11. $a \vdash b \Rightarrow a \wedge b$ МТД(10);
12. $\vdash a \Rightarrow (b \Rightarrow a \wedge b)$ МТД (11).

Нижче наведемо ряд похідних правил (основними залишаються правило MP і правило підстановки).

1. $A \Rightarrow B,\ B \Rightarrow C \vdash A \Rightarrow C$ — *правило силогізму ПС;*
2. $A \Rightarrow (B \Rightarrow C) \vdash B \Rightarrow (A \Rightarrow C)$ — *правило перестановки посилок ППП;*
3. $A \Rightarrow (B \Rightarrow C) \vdash A \wedge B \Rightarrow C$ — *правило з'єднання посилок ПЗП;*
4. $A \wedge B \Rightarrow C \vdash A \Rightarrow (B \Rightarrow C)$ — *правило роз'єднання посилок ПРП;*
5. $A, B \vdash A \wedge B$ — *правило введення кон'юнкції ВвК;*
6. $A \vdash A \vee B,\ B \vdash A \vee B$ — *правила введення диз'юнкції ВвД;*
7. $A \wedge B \vdash A,\ A \wedge B \vdash B$ — *правила вилучення кон'юнкції ВК;*
8. $\neg\neg A \vdash A$ — *правило вилучення заперечення ВЗ;*
9. $A \Rightarrow B,\ A \Rightarrow C \vdash A \Rightarrow B \wedge C$ — *правило композиції ПК;*
10. $A \Rightarrow B \vdash \neg B \Rightarrow \neg A$ — *правило контрапозиції КП;*
11. $\dfrac{\Gamma, A \vdash C, \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}$ — *правило вилучення диз'юнкції ВД;*
12. $\dfrac{\Gamma, A \vdash B; \quad \Gamma, A \vdash \neg B}{\Gamma, \vdash \neg A}$ — *правило введення заперечення або приведення до абсурду ПА (Reductio ad absurdum RA).*

Зробимо підстановку $S_{a,b,c}^{A,B,C}(2.2.5)$, отримаємо

$$\vdash (A \Rightarrow (B \Rightarrow C) \Rightarrow (B \Rightarrow (A \Rightarrow C)).$$

Отже, якщо $\vdash A \Rightarrow (B \Rightarrow C)$, то за правилом MP з останньої формули маємо $\vdash B \Rightarrow (A \Rightarrow C)$. Отримали правило перестановки посилок.

Правило з'єднання посилок (ПЗП) легко виводиться з формули (2.2.4) із застосуванням правила підстановки і MP.

Правило введення кон'юнкції (ВвК) виводиться з формули (2.3.3).

Правило роз'єднання посилок ПРП пропонуємо читачеві вивести самостійно, аналогічно до виведення попередніх правил.

Правила ВвД, ВК, ВЗ, ПК, КП, ВД безпосередньо виводяться відповідно з аксіом А6, А7, А3, А4, А11, А5, А9, А8.

Також самостійно пропонуємо вивести правило RA використовуючи нижче доведену теорему 2.3.4.

Наведемо приклади застосування похідних правил.

ТЕОРЕМА 2.3.3. $\vdash a \wedge \neg a \Rightarrow b$.

1. $a \wedge \neg a$	припущення;
2. $\vdash a$	ВК(1);
3. $\vdash \neg a$	ВК(1);
4. $\vdash a \Rightarrow (b \Rightarrow a)$	А1;
5. $\vdash a \Rightarrow (\neg b \Rightarrow a)$	$S_b^{\neg b}(4)$;
6. $\vdash \neg b \Rightarrow a$	МР(2, 4);
7. $\vdash \neg a \Rightarrow \neg\neg b$	КП;
8. $\vdash \neg\neg b$	МР(3, 7);
9. $\vdash b$	ВЗ(8);
10. $a \wedge \neg a \vdash b$	Df(1-9);
11. $\vdash a \wedge \neg a \Rightarrow b$	МТД(10).

ТЕОРЕМА 2.3.4. $\vdash (a \Rightarrow b) \Rightarrow ((a \Rightarrow \neg b) \Rightarrow \neg a)$.

1. $a \Rightarrow b$	припущення;
2. $a \Rightarrow \neg b$	припущення;
3. $\vdash a \Rightarrow b \wedge \neg b$	ПК(1, 2);
4. $\vdash \neg(b \wedge \neg b) \Rightarrow \neg a$	КП(3);
5. $\vdash a \wedge \neg a \Rightarrow b$	Т 2.3.3;
6. $\vdash b \wedge \neg b \Rightarrow \neg(a \Rightarrow \neg\neg a)$	вставити формулу;
7. $\vdash \neg\neg(a \Rightarrow \neg\neg a) \Rightarrow \neg(b \wedge \neg b)$	КП(6);
8. $\vdash a \Rightarrow \neg\neg a$	А10;
9. $\vdash (a \Rightarrow \neg\neg a) \Rightarrow \neg\neg(a \Rightarrow \neg\neg a)$	вставити формулу;
10. $\vdash \neg\neg(a \Rightarrow \neg\neg a)$	МР(8, 9);
11. $\vdash \neg(b \wedge \neg b)$	МР(10, 7);
12. $\vdash \neg a$	МР(11, 4);
13. $a \Rightarrow b, a \Rightarrow \neg b \vdash \neg a$	Df(1-12);
14. $a \Rightarrow b \vdash (a \Rightarrow \neg b) \Rightarrow \neg a$	МТД(13);
15. $\vdash (a \Rightarrow b) \Rightarrow ((a \Rightarrow \neg b) \Rightarrow \neg a)$	МТД(14).

Доведення теорем 2.3.3 та 2.3.4 — скорочені. Їх можна перетворити на повні, замінивши всі посилання на доведені раніш теореми і похідні правила, їх виведенням з аксіом за правилами підстановки і МР.

Вправи

1. Довести, що такі формули є теоремами числення висловлень:

a) $\vdash (a\Rightarrow b)\wedge(b\Rightarrow c)\Rightarrow(a\Rightarrow c)$;
b) $\vdash (a\Rightarrow(b\Rightarrow c))\Rightarrow(a\wedge b\Rightarrow c)$;
c) $\vdash (a\wedge b\Rightarrow c)\Rightarrow(a\Rightarrow(b\Rightarrow c))$;
d) $\vdash (a\Rightarrow b)\Rightarrow(a\Rightarrow(a\Rightarrow b))$;
e) $\vdash (\rceil a\Rightarrow b)\Rightarrow(c\wedge \rceil b\Rightarrow a)$;
f) $\vdash a\Rightarrow(b\Rightarrow a\wedge b)$;
g) $\vdash (a\Rightarrow b)\vee(b\Rightarrow a)$;
h) $\vdash (a\Rightarrow b)\wedge(b\Rightarrow a)$;
i) $\vdash a\wedge \rceil b\Rightarrow((a\Rightarrow b)\Rightarrow d)$;
j) $\vdash (a\vee a\Rightarrow a)\wedge(a\Rightarrow a\vee a)$.

2. Обґрунтувати вивідність:

a) $a,\ \rceil b \vdash a\Rightarrow \rceil b$;
b) $\rceil a\wedge \rceil b,\ a \vdash b$;
c) $\rceil(a\vee b),\ b \vdash a\vee b$;
d) $a\wedge \rceil b,\ a\Rightarrow b \vdash c$;
e) $a\Rightarrow b,\ a\wedge c \vdash b\wedge c$;
f) $a\Rightarrow b\vee c,\ a\wedge \rceil b \vdash c$;
g) $a\wedge b\Rightarrow c,\ \rceil c \vdash \rceil a\vee \rceil b$;
h) $a\vee b\Rightarrow c,\ \rceil c \vdash \rceil a\wedge \rceil b$;
i) $\rceil a\wedge \rceil b,\ \rceil b\Rightarrow a \vdash c$;
j) $b,\ a\Rightarrow \rceil b \vdash \rceil(\rceil b\vee a)$.

Методи встановлення вивідності:

1) побудова формального виведення;
2) доведення існування формального виведення;
3) заміна питання про вивідність питанням про логічне слідування в алгебрі висловлень.

2.4. НЕСУПЕРЕЧНІСТЬ ЧИСЛЕННЯ ВИСЛОВЛЕНЬ

Формули числення висловлень можна інтерпретувати як формули алгебри висловлень. Для цього ми будемо трактувати пропозиційні літери числення висловлень як змінні алгебри висловлень, тобто як змінні, що приймають значення 1 або 0. Пропозиційні зв'язки $\rceil$, $\wedge$, $\vee$, $\Rightarrow$ означимо аналогічно як і в алгебрі висловлень, тоді всяка формула при будь-яких значеннях змінних сама буде приймати одне із значень 1 або 0, яке обчислюється за правилами алгебри висловлень.

Проблема несуперечності виникає при розгляді будь-якого числення, це одна з кардинальних проблем математичної логіки.

ОЗНАЧЕННЯ 2.4.1. *Назвемо логічне числення несуперечним, якщо в ньому не вивідні жодні дві формули, з яких одна є запереченням другої.*

Несуперечне числення — це таке числення, в якому жодна формула A та $\rceil A$ не можуть бути одночасно виведені з аксіом цього числення за допомогою вказаних у ньому правил. Тобто одночасно не є теоремами цього числення. Це несуперечність у класичному розумінні.

Проблема несуперечності: чи є дане числення несуперечним чи ні? Якщо в теорії знаходяться вивідні формули A та $\rceil A$, то така теорія називається суперечною. Такі числення ніякої цінності не мають. Всі логічні системи такі, що якщо яка-небудь з них виявилася суперечною, то це б означало, що в ній всі формули вивідні, і тому такі системи не можуть відображати в собі різницю між істинним та хибним.

Той факт, що в суперечному численні будь-яка формула вивідна, може бути використаний для доведення несуперечності.

Для цього достатньо показати, що існує принаймні одна невивідна формула. Звідси випливає несуперечність числення. Але можна довести несуперечність простіше:

ТЕОРЕМА 2.4.1. *Кожна вивідна в численні висловлень формула, розглянута як формула алгебри висловлень, є тавтологією.*

Доведення. Легко безпосередньо перевірити, що аксіоми числення висловлень — тавтології. Покажемо, що якщо формула $A(a)$, яка містить пропозиційну змінну a, є тавтологією, то й формула $A(B) = S_a^B(A)$ теж буде тавтологією. Дійсно, $A(a)$ при всіх значеннях пропозиційної змінної приймає значення 1. Тоді $A(1)$ та $A(0)$ мають значення 1, які б не були значення інших пропозиційних змінних, але B при будь-яких значеннях пропозиційної змінної може мати значення тільки 1 або 0. Отже, зрозуміло, що $A(B)$ завжди буде мати значення 1.

Доведемо, що якщо формули A та $A \Rightarrow B$ — тавтології, то формула B теж тотожно істинна.

Якщо A — тавтологія, то вона завжди має значення 1. Оскільки формула $A \Rightarrow B$ теж завжди приймає значення 1, то B не може прийняти значення 0 на жодному наборі, інакше формула прийняла б вигляд $1 \Rightarrow 0$, яка, за означенням імплікації в алгебрі висловлень, є 0.

Отже ми показали, що 1) всі аксіоми — тавтології; 2) застосовуючи до тавтологій правила виведення, отримаємо також тавтології. Звідси випливає, що всі вивідні формули числення висловлень, які розглядаються як формули алгебри висловлень, є тавтологіями.

Отже, зрозуміло, що якщо формула A — вивідна у численні висловлень, то формула $\rceil A$ не може бути вивідна, тому що A тавтологія, а $\rceil A$ — навпаки приймає значення 0 на всіх наборах значень істинності змінних. Тому теорему формулюють так:

Числення висловлень — теорія несуперечна.

Теорема встановлює важливу властивість теорем числення висловлень. Вона дає впевненість у тому, що серед вивідних в численні висловлень формул немає жодної, відмінної від тотожньо істинної. Цю властивість називають *несуперечністю відносно істинності,* несуперечністю відносно інтерпретації.

2.5. ПОВНОТА ЧИСЛЕННЯ ВИСЛОВЛЕНЬ. ПРОБЛЕМА ВИРІШЕННЯ

Формули числення висловлень можуть бути інтерпретовані як формули алгебри висловлень. Розглядаючи несуперечність числення висловлень показали, що всяка формула, яка виводиться в численні висловлень є тавтологією, якщо її інтерпретувати як формулу алгебри висловлень. Виникає зворотне питання: чи буде кожна тотожньо істинна формула алгебри висловлень вивідною в численні висловлень?

Це питання і є проблемою повноти в широкому розумінні для числення висловлень.

Суть такої постановки питання полягає в тому, що при побудові числення, призначеного виражати змістовну логіку, необхідно знати, чи достатньо аксіом і правил виведення для того, щоб вивести будь-яку формулу, яка у змістовному розумінні є тотожно істинною.

Проблема повноти в широкому розумінні розв'язується позитивно. Введемо необхідні позначення. Буква R позначає довільну формулу, яка є вивідною в численні висловлень, а буква G позначає формулу, для якої формула $\urcorner G$ вивідна в численні висловлень.

Нехай формула A містить рівно n змінних висловлень: $A = A(a_1, ..., a_n)$. Означимо для неї за індукцією формулу:

$$\prod_{\delta_1,...,\delta_n=R,G} A(\delta_1,...,\delta_n). \qquad (*)$$

Якщо $n = 1$, то $\prod_{\delta_1=R,G} A(\delta_1)$ є формула $A(R) \wedge A(G)$.

Нехай формули (*) означені для всіх тих A, для яких $i \le k$. Нехай формула $A(a_1, ..., a_{k+1})$ містить рівно $k + 1$ пропозиційну змінну $a_1, ..., a_k, a_{k+1}$; через $\prod_{\delta_1,...,\delta_{k+1}=R,G} A(\delta_1,...,\delta_{k+1})$ позначимо формулу:

$$\prod_{\delta_1,...,\delta_k=R,G} A(\delta_1,...,\delta_k,R) \wedge \prod_{\delta_1,...,\delta_k=R,G} A(\delta_1,...,\delta_k,G).$$

Таким чином означимо формулу (*) для $A(a_1, ..., a_{k+1})$ через формули (*) для $A(a_1, ..., a_k, R)$ і $A(a_1, ..., a_k, G)$, які містять по k пропозиційних змінних.

По змісту формула (*) може бути означена як кон'юнкція всіх можливих формул, одержаних з $A(a_1, ..., a_n)$ всіма можливими замінами змінних $a_1, ..., a_n$ на R, G. Має місце таке твердження:

ЛЕМА 11.1. $\vdash \prod\limits_{\delta_1,...,\delta_n=R,G} A(\delta_1,...,\delta_n) \Rightarrow A(a_1,...,a_n)$. (2.5.1)

Це твердження доводиться за індукцією з використанням $\vdash A \wedge \neg A \Rightarrow B$ та $\vdash A(R) \wedge A(G) \Rightarrow A(a)$.

ТЕОРЕМА 2.5.1. *Кожна тотожньо істинна формула алгебри висловлень вивідна у численні висловлень.*

Для доведення розглянемо довільну тавтологію алгебри висловлень A і скористуємося формулою (2.5.1).

Оскільки формула $A = A(a_1, ..., a_n)$ — тавтологія, то будь-яка підстановка формул R, G замість $a_1, ..., a_n$ у формулу A приводить до формули, яка є вивідною у численні висловлень (доведено у [7]). Отже, у формулі $\prod\limits_{\delta_1,...,\delta_n=R,G} A(\delta_1,...,\delta_n)$, яка є посилкою в (2.5.1), всі співмножники є вивідними формулами, тоді, за ПК і вся формула є вивідною. Застосувавши до (2.5.1) правило МР, одержимо, що формула $A(a_1, ..., a_n)$ є вивідною.

Таки чином, повноту в широкому розумінні для числення висловлень доведено. Показано, що термін "вивідна у численні висловлень формула" співпадає зі змістовним поняттям тотожньо істинної формули.

Не менш важливе значення, ніж поняття повноти у широкому розумінні, має поняття повноти логічних числень у вузькому розумінні.

Логічне числення називається повними у вузькому розумінні, якщо приєднання до його аксіом якої-небудь формули, яка не є вивідною в ньому, приводить до суперечності.

Числення висловлень повне також у вузькому розумінні.

Доведення цього факту проводиться з використанням КНФ.

Проблема вирішення для формальної теорії T полягає в тому, щоб знайти метод, за допомогою якого для довільної формули A з T можна за скінчене число кроків визначити чи буде A теоремою теорії T чи ні.

У першому випадку говорять, що ця проблема розв'язується позитивно і теорію T називають розв'язною, а в другому — негативно, а теорію T — нерозв'язною.

Зауважимо, що у численні висловлень, використовуючи означення теореми числення висловлень, ще не можна з'ясувати, як розв'язується проблема вирішення для даного числення.

Але використовуючи теореми несуперечності та повноти легко розв'язати дану проблему.

ТЕОРЕМА 2.5.2. *Проблема вирішення для числення висловлень розв'язується позитивно.*

Доведення. Нехай A — довільна формула числення висловлень. Застосуємо до неї один з алгоритмів перевірки A на тавтологію. Складемо для A таблицю істинності і розглянемо її останній стовпець. Це можна зробити за скінчене число кроків. Якщо з'ясується, що A — тавтологія (останній стовпець складається з одних лише 1), то за теоремою про повноту A — теорема, якщо A не є тавтологія (в останньому стовпці міститься хоча б один 0), то за теоремою про несуперечність (її контрапозицією) A не є теоремою.

2.6. НЕЗАЛЕЖНІСТЬ АКСІОМ ЧИСЛЕННЯ ВИСЛОВЛЕНЬ

Як вже зазначалось, будь-яке логічне числення може бути задане в такий спосіб: дається означення поняття формули і поняття вивідної формули. Це робиться шляхом завдання:

1) деяких вихідних формул, оголошених вивідними, що називаються аксіомами, та,

2) правил виведення, тобто правил, за допомогою яких з вивідних формул можна утворити нові вивідні формули.

Для кожного такого числення виникає питання про незалежність його аксіом. А саме:

Чи можна яку-небудь аксіому вивести з решти, застосовуючи правила виведення даної системи? Якщо деяку аксіому можна вивести з решти, то її можна викреслити із списку аксіом і логічне числення при цьому не зміниться, тобто запас вивідних формул зостанеться тим же.

ОЗНАЧЕННЯ 2.6.1. *Аксіома, яка не виводиться з інших аксіом системи називається незалежною від цих аксіом, а система аксіом, в якій жодна з аксіом не виводиться з інших, називається незалежною системою аксіом. В іншому випадку — залежною системою аксіом.*

Питання про незалежність однієї аксіоми деякої системи від інших аксіом часто буває рівносильне питанню про можливість заміни без суперечностей у розглядуваній системі дану аксіому її запереченням. Наприклад: питання про незалежність п'ятого постулату Евкліда в системі аксіом геометрії.

Для розв'язання питання про незалежність деякої аксіоми A числення висловлень, ми поставимо собі завдання інтерпретувати пропозиційні літери, як змінні із скінченим набором значень, які ми будемо позначати грецькими літерами α, β, ... Операції означимо так, щоб виконувались такі умови:

1) всі аксіоми, крім аксіоми A при всіх значеннях змінних приймають значення α;

2) кожна формула, яка є вивідною з сукупності всіх відмінних від A аксіом системи, також приймає значення α при всіх значеннях вхідних змінних;

3) аксіома A приймає відмінні від α значення при деяких значеннях вхідних змінних.

Зрозуміло, що якщо вдасться привести таку інтерпретацію, то незалежність аксіоми A від інших аксіом доведена. Оскільки, якщо б A була вивідною з них, то вона б мала значення α при всіх значеннях змінних.

Зауважимо, що формули, в яких замість змінних поставлені деякі їх значення, також мають зміст.

Наприклад: $\alpha \wedge \beta$, $\rceil \alpha$, $\beta \Rightarrow \gamma$, ...

Той факт, що дві формули A і B приймають при всіх замінах змінних, які входять в них, однакові значення α, β, ... позначатимемо: $A = B$. Знак = зв'язує слабше логічних зв'язок $\rceil$, $\wedge$, $\vee$, $\Rightarrow$.

Доведемо незалежність аксіоми А3.

Пропозиційні літери інтерпретуємо як змінні, що приймають два значення α і β.

Пропозиційні зв'язки (крім $\wedge$) означимо як в алгебрі висловлень:

$$\alpha \Rightarrow \alpha = \alpha; \qquad \beta \Rightarrow \beta = \alpha; \qquad \beta \Rightarrow \alpha = \alpha; \qquad \alpha \Rightarrow \beta = \beta;$$
$$\alpha \vee \alpha = \alpha; \qquad \alpha \vee \beta = \alpha; \qquad \beta \vee \alpha = \alpha; \qquad \beta \vee \beta = \beta;$$
$$\rceil \alpha = \beta; \qquad \rceil \beta = \alpha.$$

Пропозиційну зв'язку $\wedge$ означимо умовою: $a \wedge b = b$.

Покажемо, що всі формули А1 – А11 крім А3 приймають значення α при всіх можливих значеннях пропозиційних змінних. У формули груп I, II, IV "$\wedge$" не входить, а інші операції означені, як в алгебрі висловлень. Оскільки в алгебрі висловлень ці формули є тотожньо істинними, то в нашій інтерпретації вони приймають значення α при будь-яких значеннях змінних.

Розглянемо формули групи II. Формула А4 приймає значення α, оскільки в нашій інтерпретації вона рівносильна $b \Rightarrow b$. Формула А5 рівносильна формулі

$$(a \Rightarrow b) \Rightarrow ((a \Rightarrow c) \Rightarrow (a \Rightarrow c)).$$

Ця формула не містить кон'юнкції і є тотожньо істинною формулою алгебри висловлень, тому завжди приймає значення α.

Але формула А3 тотожньо не дорівнює α. Дійсно, при $a = \beta$, $b = \alpha$, вони приймають вигляд: $\beta\wedge\alpha \Rightarrow \beta$, але за означенням $\beta\wedge\alpha = \alpha$ і формула приймає вигляд $\alpha \Rightarrow \beta$, а цей вираз, за умовою, має значення β.

Тепер покажемо, що формули, які одержуються шляхом застосування правил виведення з таких, які тотожньо рівні α, самі тотожньо дорівнюють α.

Для правила підстановки це очевидно: якщо формула при всіх значеннях змінної приймає значення α, то такою ж буде і формула одержана з неї будь-якою заміною змінних.

Розглянемо правило МР. Нехай формули A та $A \Rightarrow B$ приймають значення α при всіх значеннях змінних, що в них входять, тоді $A \Rightarrow B = \alpha \Rightarrow B$.

У цьому випадку B не може приймати значення β, інакше одержимо $A \Rightarrow B = \alpha \Rightarrow \beta = \beta$, чого не може бути.

Тобто незалежність А3 доведена.

Взагалі незалежність будь-якої аксіоми з груп II–IV можна довести за такою схемою. Припускаємо, що змінні можуть приймати тільки два значення α і β. Всі логічні операції $\rceil$, $\wedge$,$\vee$, $\Rightarrow$, крім однієї з них, ми означаємо так само як і в алгебрі висловлень, причому α є 1, β є 0. Одну ж з операцій означаємо так, щоб та аксіома, незалежність якої доводиться, не була тотожньо рівна α.

Наведемо таблицю, у першому стовпці якої стоїть аксіома, незалежність якої доводиться, у другому — дано означення тієї операції $\rceil$, $\wedge$, $\vee$, $\Rightarrow$, яка означається інакше, ніж в алгебрі висловлень, а у третьому стовпці вказуються ті значення змінних, при яких відповідна аксіома приймає значення β.

А3	$a \wedge b \Rightarrow a$	$a\wedge b = b$	$a = \beta;\ b = \alpha$
А4	$a \wedge b \Rightarrow b$	$a\wedge b = a$	$a = \alpha;\ b = \beta$
А5	$(a \Rightarrow b)\Rightarrow ((a \Rightarrow c)\Rightarrow (a \Rightarrow b\wedge c))$	$a\wedge b = \beta$	$a=\alpha;\ b=\alpha;\ c=\alpha$
А6	$a \Rightarrow a\vee b$	$a\vee b = b$	$a = \alpha;\ b = \beta$
А7	$b \Rightarrow a \vee b$	$a\vee b = a$	$a = \beta;\ b = \alpha$
А8	$(a \Rightarrow c) \Rightarrow ((b \Rightarrow c) \Rightarrow (a\vee b \Rightarrow c))$	$a\vee b = \alpha$	$a=\beta;\ b=\beta;\ c=\beta$
А9	$(a \Rightarrow b) \Rightarrow (\rceil b \Rightarrow \rceil a)$	$\rceil a = a$	$a = \beta;\ b = \alpha$
А10	$a \Rightarrow \rceil\rceil a$	$\rceil a = \beta$	$a = \alpha$
А11	$\rceil\rceil a \Rightarrow a$	$\rceil a = \alpha$	$a = \beta$

При всіх вказаних інтерпретаціях аксіоми, що входять у групи відмінні від тієї, в якій знаходиться досліджувана аксіома, приймають значення α при всіх значеннях змінних. Оскільки в аксіоми цих груп не входить та виключна операція, яка означається інакше, ніж в алгебрі висловлень, то інтерпретація цих формул така ж як і в алгебрі висловлень. Тому ці формули приймають значення α при будь-яких значеннях змінних.

Для аксіом тієї групи, куди входить досліджувана аксіома, можна переконатись безпосередньо з допомогою перевірки, що дві з них також тотожньо рівні α, а досліджувана аксіома приймає значення β при значеннях змінних вказаних у третьому стовпці.

Доведення того, що правила виведення, які застосовуються до формул тотожньо рівних α, породжують формули також тотожньо рівні α для всіх інтерпретацій, аналогічне доведеному раніше.

Покажемо незалежність аксіом групи I.

Інтерпретації задовольняють наступним загальним умовам:

$$\begin{array}{lll} a \Rightarrow a = \alpha; & a \Rightarrow \alpha = \alpha; & \beta \Rightarrow a = \alpha; \\ a \wedge b = b \wedge a; & a \wedge \alpha = a; & a \wedge \beta = \beta; \\ a \vee b = b \vee a; & a \vee \alpha = \alpha; & a \vee \beta = a; \\ \neg\alpha = \beta;\ \neg\beta = \alpha; & a \wedge a = a; & a \vee a = a. \end{array} \qquad (a)$$

Ці умови сумісні, оскільки їм задовольняє алгебра висловлень.

Для доведення незалежності A1 виберемо таку інтерпретацію. Змінні приймають значення $\alpha, \beta, \gamma, \delta$ і крім (a) виконуються такі умови:

$$\begin{array}{lll} \alpha \Rightarrow \beta = \beta; & \alpha \Rightarrow \gamma = \beta; & \alpha \Rightarrow \delta = \beta; \\ \gamma \Rightarrow \beta = \beta; & \gamma \Rightarrow \delta = \beta; & \\ \delta \Rightarrow \beta = \beta; & \delta \Rightarrow \gamma = \alpha; & \\ \gamma \wedge \delta = \delta; & \gamma \vee \delta = \gamma; & \neg\gamma = \delta;\ \neg\delta = \gamma. \end{array} \qquad (b)$$

Умови (a) та (b) повністю визначають інтерпретацію. З них випливає, що застосування MP до формули, яка тотожно дорівнює α, приводить до формули, яка тотожно дорівнює α.

Дійсно, якщо $A = \alpha$ і $A \Rightarrow R = \alpha$, то $\alpha \Rightarrow R = \alpha$, але з ($a$) і ($b$) випливає, що коли $|R| \neq \alpha$, то $\alpha \Rightarrow R \neq \alpha$, тому R також тотожньо дорівнює α.

Підстановка у формулу, яка тотожно дорівнює α, приводить до формули, що дорівнює α. Отже, правила виведення застосовані до формул, які тотожно дорівнюють α, приводять до формул тотожно рівних α.

У розглядуваній інтерпретації A1 при значеннях змінних $a = \delta$, $b = \alpha$. приймає значення β:

$$\delta \Rightarrow (\alpha \Rightarrow \delta) = \delta \Rightarrow \beta = \beta.$$

Решта аксіом при всіх значеннях змінних приймає значення α. Розглянемо, наприклад, A2:

$$(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c)).$$

Покажемо, що коли $a \Rightarrow b = \alpha$, $b \Rightarrow c = \alpha$, то $a \Rightarrow c = \alpha$ і $A2 = \alpha$. Дійсно, якщо $a \Rightarrow b = \alpha$, то можливі випадки: 1) $a = \beta$; 2) $b = \alpha$; 3) $a = b$; 4) $a = \delta$, $b = \gamma$.

У перших трьох випадках безпосередньо видно, що $a \Rightarrow c = \alpha$, оскільки при $b = \alpha$ з $b \Rightarrow c = \alpha$ випливає $c = \alpha$. В останньому випадку, в силу того, що $b \Rightarrow c = \alpha$, для c візьмемо два значення: γ і α, в обох випадках $a \Rightarrow c = \alpha$.

Залишається випадок коли, або $a \Rightarrow b$, або $b \Rightarrow c$ не рівне α. Зауважимо, що імплікація може приймати значення тільки α або β (легко бачити з (*a*) і (*b*)). Якщо $\alpha \Rightarrow b = \beta$, то $(a \Rightarrow b) \Rightarrow (a \Rightarrow c) = \alpha$.

Якщо $b = c = \beta$ і a приймає значення відмінні від β , то $a \Rightarrow (b \Rightarrow c) = \beta$ і вся формула A2 приймає значення α.

Якщо ж $a = \beta$, то $a \Rightarrow c = \alpha$ і $A2 = \alpha$. Отже, $|A2| = \alpha$ при будь-яких значеннях змінних.

Доведемо незалежність A2. Для цього виберемо інтерпретацію в якій змінні приймають три значення α, β, γ.

Операції означимо умовами (*a*) і додамо ще такі:

$$\alpha \Rightarrow \beta = \beta,\ a \Rightarrow \gamma = \gamma,\ \gamma \Rightarrow \beta = \gamma,\ \neg\gamma = \gamma. \qquad (c)$$

Умовами (*a*) і (*c*) операції імплікації і заперечення означені сповна.

Операції $\wedge$ і $\vee$ в цьому випадку повністю означені умовами (*a*). Дійсно, якщо одна із змінних мас значення α або β, то застосовуємо умови (*a*), якщо обидві змінні мають значення γ, то $a \wedge b = \gamma$.

У цій інтерпретації A2 при $a = \gamma$, $b = \gamma$, $c = \beta$ одержимо $(\gamma \Rightarrow (\gamma \Rightarrow \beta)) \Rightarrow ((\gamma \Rightarrow \gamma) \Rightarrow (\gamma \Rightarrow \beta)) = (\gamma \Rightarrow \gamma) \Rightarrow (\alpha \Rightarrow \gamma) = \alpha \Rightarrow \gamma = \gamma$, тобто A2 приймає значення відмінні від α.

Решта аксіом приймають значення α при будь-яких значеннях змінних.

Наприклад A1: $a \Rightarrow (b \Rightarrow a)$.

При $a = b$ формула приймає значення α. У випадку $a = \alpha$ маємо $b \Rightarrow \alpha = \alpha$ і $A1 = \alpha$. При $a = \beta$ також $A1 = \alpha$. Залишається $a = \gamma$. Тоді $\gamma \Rightarrow (b \Rightarrow \gamma)$, але $b \Rightarrow \gamma$ може приймати значення α або γ, в обох випадках $\gamma \Rightarrow (b \Rightarrow \gamma) = \alpha$.

Розглянемо також A6: $a \Rightarrow a \vee b$.

Якщо $a = \beta$, то $A6 = \alpha$.

Якщо $a = \alpha$, то $a \vee b = \alpha$ і $A6 = \alpha$.

Нехай $a = \gamma$, тоді

$$\gamma \Rightarrow \gamma \vee b. \qquad (*)$$

Якщо $b = \alpha$ або $b = \gamma$, то вираз (*) рівний α.

Якщо ж $b = \beta$, то $\gamma \vee b = \gamma$ і формула $A6 = \alpha$.

Розглянемо ще А9: $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$.

У випадку, коли a і b рівні α або β, то трактуючи α — 1, β — 0 одержимо, що А9 приймає значення α, оскільки тоді мають місце закони алгебри висловлень.

Якщо $a = b$, то А9 рівна $\alpha \Rightarrow \alpha = \alpha$. Припустимо, що $a = \gamma$, одержимо $(\gamma \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg \gamma)$, але $\neg \gamma = \gamma$, отже

$$(\gamma \Rightarrow b) \Rightarrow b(\neg b \Rightarrow \gamma). \qquad (+)$$

Якщо $b = a$, то $\neg b = \beta$, $\neg b \Rightarrow \gamma = \alpha$ і вираз (+) рівний α.

Якщо $b = \beta$, то $(\gamma \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma) = \gamma \Rightarrow \gamma = \alpha$.

Якщо $b = \gamma$, $a = \alpha$, то $(\alpha \Rightarrow \gamma) \Rightarrow (\gamma \Rightarrow \beta) = \gamma \Rightarrow \gamma = \alpha$.

Якщо $b = \gamma$, $a = \beta$, то $(\beta \Rightarrow \gamma) \Rightarrow (\gamma \Rightarrow \alpha) = \alpha \Rightarrow \alpha = \alpha$.

Отже, А9 приймає значення α при всіх значеннях змінних.

Аналогічно доводиться теж саме для решти аксіом.

Таким чином доведена така теорема.

ТЕОРЕМА 2.6.1. *Система аксіом числення висловлень є незалежною.*

Доведена властивість незалежності аксіом числення висловлень не має такого принципового значення, як доведення несуперечності чи повноти. Справді, залежність системи аксіом означає тільки, що є зайві аксіоми, а це теорії не загрожує.

Розділ 3

ЛОГІКА ПРЕДИКАТІВ

3.1. ПОНЯТТЯ ПРЕДИКАТА

Числення висловлень дозволяє формалізувати лише малу частину множини міркувань. Існують такі логічні міркування, які не можуть бути обґрунтовані в рамках логіки висловлень, побудованої як алгебра висловлень або як аксіоматична теорія. Розглянемо, наприклад, такі висловлення:

(I) Кожний друг Михайла є другом Івана;
Петро не є другом Івана,
отже, Петро не є другом Михайла.

(II) Всі люди смертні;
Сократ — людина,
отже, Сократ смертний.

Позначимо речення "Кожний друг Михайла є другом Івана" — p, речення Петро не є другом Івана" — q, речення "Петро не є другом Михайла" — r, тоді (I) можемо записати так:

$$p \wedge q \to r. \tag{3.1.1}$$

Аналогічно для речення (II), якщо позначити “Всі люди смертні” — p, “Сократ — людина” — q, “Сократ смертний” — r, то отримаємо формулу (13.1).

При значеннях істинності $|p| = |q| = 1$, $|r| = 0$ формула (3.1.1) приймає значення істинності 0, хоча (I) і (II) — очевидно істинні. Довести це в логіці висловлень неможливо, оскільки в розглядуваних міркуваннях для отримання з посилок висновку використовується внутрішня структура висловлень p, q, r.

Отже, для вивчення навіть найпростіших міркувань необхідний аналіз внутрішньої структури висловлень.

Проте такий аналіз не можливий на базі алгебри висловлень, оскільки в цій теорії висловлення виступає як вихідний пункт дослідження, як неподільний об’єкт, позбавлений частин та внутрішньої структури.

Однак зрозуміло, що у природних мовах висловлення має внутрішню структуру. Інакше кажучи, значення висловлення, хоча б у першому наближенні, є функцією значень його компонентів.

До того як буде розглянутий семантичний аспект, навіть простий лексичний аналіз показує цікаву схожість між висловленнями у кожному з наведених міркувань.

Наприклад, у міркуванні (II) деякі слова присутні більше одного разу. Отже, можна припустити, що вказана схожість повинна зберігатися у формальній версії цього міркування.

З іншого боку, у природній мові часто опускають легко відновлювані слова. Текст (II), в цьому відношенні, не є виключенням. Ми вже помітили наявність зв'язки кон'юнкції, позначеної у тексті крапкою з комою. Перше висловлення містить також приховану імплікацію. Його можна переписати таким чином:

Бути людиною означає бути смертним.

Це переформулювання не зовсім задовільне: складається таке враження, що йому відповідає формула типу $a \to b$. Але це не так: слово "бути" встановлює обумовлений зв'язок між посилкою a та висновком b. Зауважимо, що слово "Сократ" встановлює аналогічний зв'язок між останніми рядками в (II). Проте, ці випадки відмінні: слово "бути" означає якийсь, поки не означений, член деякого універсуму. За аналогією з мовою математики говорять, що "бути" — це змінна (заново позначимо її як це прийнято, через x), тоді як "Сократ" — константа.

Текст (II) можна переписати так, що буде краще видно його реальну структуру, а саме таким чином:

Для всіх x, якщо x є людиною, то x є смертним;
Сократ є людиною,
(отже) Сократ є смертним.

Тепер можна поповнити запас різних складових. В доповнення до висловлень та зв'язок (кон'юнкції, диз'юнкції, імплікації) у нас є нові компоненти: “для всіх”, “x”, “Сократ”, “є людиною” і “є смертним”. Логіка предикатів є розвитком алгебри висловлень. Але логіка предикатів розглядає висловлення у відношенні до предметів. Внутрішню структуру висловлень називають суб'єктно — предикатною. Суб'єкт — назва предмета, предикат — назва властивості предмета або відношення між предметами, про які йдеться у висловленні.

У суб'єктно-предикатній структурі висловлення вихідним пунктом є поняття предиката. Для символічного позначення предикатів вживатимемо великі (прописні) латинські літери, а для позначення суб'єктів малі латинські літери.

Нехай — A довільна непорожня множина, а x — деякий предмет з A. Тоді $F(x)$ означає висловлення, яке стає визначеним, коли x замінено назвою певного предмета з A.

Наприклад, нехай A це множина натуральних чисел і $F(x)$ — "x є просте число", $H(x,y)$: "x менше y". Оскільки кожне висловлення є істинним або хибним, то вираз $F(x)$ означає, що кожному предмету з A поставлений у відповідність один з символів: 1 або 0. Отже, $F(x)$ на множині натуральних чисел визначає логічну функцію, яка для значень аргументу 1, 2, 3, 4, 5, ... набуває відповідно значень 0, 1, 1, 0, 1, ...

Аналогічно, невизначені висловлення $H(x, y)$, $G(x, y, z)$ є функціями двох, трьох змінних. Ці невизначені висловлення, або функції однієї чи декількох змінних, ми будемо називати *логічними функціями* або *предикатами*.

ОЗНАЧЕННЯ 3.1.1. *Будемо називати n-місним предикатом, визначеним на множині A, речення, яке містить змінні x_1, x_2, ..., x_n та яке перетворюється у висловлення при підстановці замість цих змінних назв будь-яких конкретних елементів з множини A.*

Для n-місного предиката будемо використовувати позначення: $P(x_1, x_2, ..., x_n)$. Змінні $x_1, x_2, ..., x_n$ називають *предметними*, а елементи множини A, які ці змінні пробігають, — *конкретними предметами*.

Відмітимо ще один підхід до поняття предиката. Як зазначалось, предикат $P(x_1, ..., x_n)$, визначений на множині A, перетворюється у конкретне висловлення $P(a_1, ..., a_n)$, якщо замість предметних змінних $x_1, ..., x_n$ підставити в нього назви конкретних предметів (елементів $a_1, ..., a_n$) з множини A. Це висловлення може бути або істинним, або хибним, тобто його логічне значення дорівнює 1 або 0. Отже даний предикат визначає функцію n-аргументів, задану на множині A і яка набуває значень на двоелементній множині $\{0, 1\}$. Іноді цю функцію і називають предикатом.

Предикатом з однією змінною можна виразити властивість предмета: $P(x)$ — "x — просте число", $T(x)$ — "x — прямокутний трикутник". Предикатами з декількома змінними можна виразити відношення між предметами: $H(x, y)$ — "x менше y", $S(x, y) - x + y + 2 = 0$.

Отже, при $n = 1$ предикат називається "властивістю", при $n > 1$ "відношенням". Елементарні висловлення можна розглядати як нульмісні предикати.

Поняття предиката у класичній логіці Аристотеля відповідає предикату з однією змінною в сучасній математичній логіці. Введення предикатів від декількох змінних, здатних відображати відношення між предметами, привносить істотно нове у порівнянні з логікою предикатів від однієї змінної. Як з'ясувалось, у всіх системах аксіом математичних дисциплін існують аксіоми, які не можна зобразити безпосередньо предикатами від однієї змінної. Проте, за допомогою предикатів від більшого числа змінних, всі аксіоми цих систем можуть бути зображеними.

Іноді доцільно виділити так звану універсальну множину (універсум), для якої всі розглядувані множини будуть підмножинами.

ОЗНАЧЕННЯ 3.1.2. *Множина M — предметна область або просто область. Елементи цієї області позначають малими латинськими літерами. Літери кінця латинського алфавіту x, y, z,* x_1*,* x_2*, ... — позначають невизначені елементи області, їх називають предметними змінними. Літерами початку латинського алфавіту — a, b, c, ...,* a_1 *,* a_2 *,... — позначають певні елементи області. Їх називають індивидні предмети або предметні константи.*

ОЗНАЧЕННЯ 3.1.3. *A, B, ..., X,* A_1*,* A_2*, ...- змінні, які приймають значення 1 або 0, їх називають змінні висловлення. Вирази F(x), G(x,y),* $P(x_1, ..., x_n)$*, ... — називають предикатними змінними, причому* $P(x_1, ..., x_n)$ *— n-місна предикатна змінна.*

ОЗНАЧЕННЯ 3.1.4. *Два n-місні предикати* $P(x_1, ..., x_n)$ *та* $Q(x_1, ..., x_n)$*, що задані на множині M, називаються рівносильними, якщо n-місний набір* $a_1, ..., a_n$ *предметних констант з M перетворює перший предикат в істинне висловлення* $P(a_1, ..., a_n)$ *тоді і тільки тоді, коли цей набір перетворює в істинне висловлення* $Q(a_1, ..., a_n)$ *— другий предикат.*

Оскільки значеннями предикатів є висловлення, то предикати можна зв'язувати знаками логічних операцій алгебри висловлень: заперечення – $\rceil$, кон'юнкція – $\wedge$, диз'юнкція — $\vee$, імплікація – $\Rightarrow$, еквіваленція – $\Leftrightarrow$.

ОЗНАЧЕННЯ 3.1.5. *Запереченням n-місного предиката* $P(x_1, x_2, ... , x_n)$*, визначеного на множині M називається новий n-місний предикат, визначений на тій же множині, який перетворюється в істинне висловлення при всіх тих і тільки тих значеннях предметних змінних, при яких вихідний предикат перетворюється в хибне висловлення.*

Позначається $\rceil P(x_1, x_2, ..., x_n)$, читається "не правильно, що $P(x_1, x_2, ..., x_n)$".

ОЗНАЧЕННЯ 3.1.6. *Кон'юнкцією n-місного предиката* $P(x_1, x_2, ..., x_n)$ *і m-місного предиката* $Q(y_1, y_2, ..., y_m)$*, визначених на множині M, називається новий (n + m)-місний предикат, визначений на множині M, який перетворюється в істинне висловлення при всіх тих і тільки тих значеннях предметних змінних, при яких обидва вихідних предикати перетворюються в істинне висловлення.*

Позначається $P(x_1, ..., x_n) \wedge Q(y_1, y_2, ..., y_m)$ — читається "$P(x_1, x_2, ..., x_n)$ і $Q(y_1, y_2, ..., y_m)$".

Операцію кон'юнкція можна застосувати до предикатів, які мають спільні змінні. В цьому випадку число змінних у новому предикаті дорівнює числу $n + m - k$, де n — число змінних першого предиката, m — число змінних другого предиката, k — число змінних, спільних для обох предикатів.

ОЗНАЧЕННЯ 3.1.7. *Диз'юнкцією n-місного предиката $P(x_1, ..., x_n)$ і m-місного предиката $Q(y_1, y_2, ..., y_m)$, визначених на множині M називається новий $(n + m)$-місний предикат, визначений на множині M, який перетворюється в істинне висловлення при всіх тих і тільки тих значеннях предметних змінних, при яких в істинне висловлення перетворюється принаймні один із вихідних предикатів.*

Позначається $P(x_1, x_2, ..., x_n) \vee Q(y_1, y_2, ..., y_m)$, читається "$P(x_1, x_2,..., x_n)$ або $Q(y_1, y_2, ..., y_m)$".

Приклад 3.1.1. Диз'юнкцією одномісних предикатів "$x \neq 0$" та "$y \neq 0$", визначених на R, є двомісний предикат $(x \neq 0) \vee (y \neq 0)$, також визначений на R, який рівносильний предикату "$x^2 + y^2 \neq 0$" на R.

Імплікація $(P(x_1, x_2, ..., x_n) \Rightarrow Q(y_1, y_2, ..., y_m))$ означається як такий предикат, що для будь-яких предметів $a_1, a_2, ..., a_n, b_1, ..., b_m$ з множини M висловлення $(P(a_1, a_2, ..., a_n) \Rightarrow Q(b_1, ..., b_m))$ є імплікацією висловлень $P(a_1, a_2, ..., a_n)$ і $Q(b_1, b_2, ..., b_m)$. Неважко перевірити, що імплікація двох предикатів, що залежать від одних і тих же змінних, хибна тоді і тільки тоді, коли посилка істинна і висновок хибний. Аналогічно означається еквіваленція двох предикатів. Еквіваленція істинна, тоді і тільки тоді, коли вихідні предикати приймають однакові значення істинності.

Нехай M — множина, на якій визначені предикати. Предикату $P(x_1, x_2, ..., x_n)$ можна поставити у відповідність множину тих елементів a_i з M, для яких висловлення $P(a_1, a_2, ..., a_n)$ — істинне.

Множиною істинності предиката $P(x_1, x_2 ,..., x_n)$, заданого на множині M, назвемо сукупність всіх впорядкованих n-ок $(a_1, a_2, ..., a_n)$ таких, що даний предикат перетворюється в істинне висловлення $P(a_1, a_2, ..., a_n)$ при підстановці $x_1 = a_1, x_2 = a_2, ..., x_n = a_n$. Позначатимемо цю множину E_P.

Множина E_P істинності n-місного предиката $P(x_1, ..., x_n)$ є n-арним відношенням між елементами множини M. Якщо предикат $P(x)$ однієї змінної, заданий на множині M, то його множина істинності E_P є підмножиною множини M.

Приклад 3.1.2. Множиною двомісного предиката "Точка x належить прямій y", заданого на множині M всіх точок площини, є бінарне відношення належності між точками і прямими площини.

Приклад 3.1.3. Множиною двомісного предиката $S(x, y)$: "$x^2 + y^2 = 9$", заданого на множині R, є множина всіх таких пар дійсних чисел, які є координатами точок площини, що утворюють коло з центром у початку координат, радіуса 3. Нарешті, якщо $A(x)$: "$|x| > 2$" — одномісний предикат на R, то $E_A = (-\infty, -2) \cup (2, +\infty)$.

Нехай $A(x) = D(x) \vee C(x)$, тоді $E_A = E_D \cup E_C$. Дійсно, якщо $x \in E_A$, то $A(x)$ — істинне, отже $D(x)$ або $C(x)$ істинне. Тобто $x \in E_D$ або $x \in E_C$, а це означає, що $x \in E_D \cup E_C$. Навпаки, нехай $x \in E_D \cup E_C$. Тоді $x \in E_D$ або $x \in E_C$, тобто $D(x)$ — істинне або $C(x)$ — істинне. Отже, $A(x)$ — істинне, тому $x \in E_A$.

У випадку, коли $A(x) = D(x) \wedge C(x)$, то $E_A = E_D \cap E_C$. Множина, відповідна предикату $\rceil P(x)$, є доповненням до множини, яка відповідає предикату $P(x)$: $E_{\rceil P} = CE_P$.

Аналогічним чином можна встановити зв'язок між множинами і предикатами від декількох змінних.

Введені вище операції над предикатами в певному розумінні аналогічні відповідним операціям над висловленнями. Специфіка природи предикатів дозволяє ввести над ними такі операції, які не мають аналогів серед операцій над висловленнями. Будемо вживати ще дві нові логічні операції. Операції ці виражають собою твердження загальності та існування і називаються операціями квантифікації.

Нехай $P(x)$ — предикат, який приймає значення 1 або 0 для кожного елемента x деякої області M.

ОЗНАЧЕННЯ 3.1.8*. Операцією зв'язування квантором загальності називається правило, за яким кожному одномісному предикату $P(x)$ ставиться у відповідність $\forall x(P(x))$ — висловлення, яке істинне, коли $P(a)$ істинне для кожного елемента $a \in M$, і хибне в іншому випадку. Це висловлення вже не залежить від x. Символ $\forall x$, називають квантором загальності. Його можна читати "для кожного x", "для довільного x", "для всіх x".*

Символ $\forall$ походить від першої літери англійського слова "All" — все.

Якщо одномісний предикат заданий на скінченній множині $M = \{a_1, a_2, ..., a_m\}$, то неважко зрозуміти, що висловлення $\forall x(P(x))$ рівносильне (має те саме логічне значення) кон'юнкту $P(a_1) \wedge P(a_2) \wedge ... \wedge P(a_m)$. Дійсно, за означенням 3.1.8 істинність висловлення $\forall x P(x)$ означає, що предикат тотожно істинний, тобто кожне з висловлень $P(a_1)$, $P(a_2)$, ..., $P(a_m)$, в які цей предикат перетворюється, істинне. Останнє рівносильне істинності кон'юнкта $P(a_1) \wedge P(a_2) \wedge ... \wedge P(a_m)$.

Отже, для предикатів, що задані на скінченній множині, операція зв'язування квантором загальності може бути виражена через кон'юнкцію. Для предикатів, заданих на нескінченній множині, такого зробити неможливо, і у цьому випадку операція зв'язування квантором загальності є істотно новою.

ОЗНАЧЕННЯ 3.1.9. *Операцією зв'язування квантором існування називається правило, за яким кожному одномісному предикату $P(x)$ ставиться у відповідність $\exists x\ (P(x))$ — висловлення, яке істинне, якщо знайдеться елемент a області M, для якого $P(a)$ істинне висловлення, і хибне в протилежному випадку.*

Знак $\exists x$ називають квантором існування по змінній x. Його можна читати "існує x", "знайдеться x". Символ $\exists$ походить від першої літери англійського слова "Exist" — існувати.

Квантори $\forall$ і $\exists$ називають *двоїстими* один одному.

Якщо одномісний предикат заданий на скінченій множині $M = \{a_1, a_2, ..., a_m\}$, то висловлення $\exists x(P(x))$ еквівалентне (має те саме логічне значення) диз'юнкту $P(a_1) \vee P(a_2) \vee ... \vee P(a_m)$. Дійсно, за означенням 3.1.9 хибність висловлення $\exists x P(x)$ означає, що предикат тотожно хибний, тобто кожне з висловлень $P(a_1)$, $P(a_2)$, ..., $P(a_m)$, в які цей предикат може перетворитися, хибне. Останнє рівносильне хибності диз'юнкта $P(a_1) \vee P(a_2) \vee ... \vee P(a_m)$.

Отже, для предикатів, що задані на скінченій множині, операція зв'язування квантором існування може бути виражена через диз'юнкцію. Для предикатів, заданих на нескінченній множині, такого зробити неможливо, і у цьому випадку операція зв'язування квантором існування є істотно новою.

Важливе значення в логіці предикатів має поняття області дії квантора (ОДК) — це вираз до якого відноситься даний квантор. ОДК обмежують дужками. Початок області дії позначають лівою дужкою, яка ставиться безпосередньо після кванторної змінної даного квантора, а відповідна їй права дужка позначає кінець ОДК.

Наведемо приклади (область дії квантора підкреслено).

$$\exists x\underline{((x \,\vdots\, 2) \Rightarrow (x \,\vdots\, 4))}; \tag{3.1.2}$$

$$\exists x\underline{(x \,\vdots\, 2)} \Rightarrow (x \,\vdots\, 4); \tag{3.1.3}$$

$$\forall x(\underline{x^2 - 5x + 6 = 0}) \Rightarrow x = 1; \tag{3.1.4}$$

$$\forall x(\underline{x^2 - 5x + 6 = 0 \Rightarrow x = 1}). \tag{3.1.5}$$

Тут універсальна множина — множина всіх дійсних чисел R.

Вирази (3.1.2) і (3.1.3), а також (3.1.4) і (3.1.5) відрізняються лише областю дії квантора, проте відмінність між ними дуже істотна.

ОЗНАЧЕННЯ 3.1.10. *Входження предметної змінної x у вираз W називають зв'язаним, якщо вона зв'язана квантором по змінній x, тобто x знаходиться в області дії квантора або безпосередньо слідує за квантором. В іншому випадку входження x у вираз W називають вільним.*

ОЗНАЧЕННЯ 3.1.11. *Предметна змінна x називається зв'язаною (вільною) в W, якщо існує зв'язане (вільне) входження x в W.*

У виразах (3.1.2), (3.1.5) входження змінної x зв'язане.

У виразах (3.1.3), (3.1.4) перше і друге входження змінної x — зв'язане, а третє — вільне; змінна x, за означенням, є і вільною, і зв'язаною.

Можна відмітити ще одну особливість операції зв'язування кванторами у порівнянні з попередніми операціями. Ті операції ставили у відповідність одному або двом предикатам новий предикат, а операція зв'язування кванторами ставить у відповідність одномісному предикату висловлення, а n-місному предикату — (n-1)-місний предикат.

Приклад 3.1.4. Розглянемо двомісний предикат "$x < y$", визначений на множині N. Застосуємо до нього квантор загальності по змінній y. Отримаємо одномісний предикат $\forall y(x < y)$, який залежить від змінної x. Цей предикат може перетворюватись як у істинне висловлення (при $x = 1$), так і у хибне (при підстановці замість x будь-яких натуральних чисел, крім 1).

Зв'язність змінної x у виразі W характеризується властивостями:

1) вираз W від x не залежить, заміна x на іншу змінну, що не входить у W, не змінює значення W;

2) підставляти в W замість x назви предметів не має сенсу.

Вправи

1. З'ясувати, які з наступних виразів є предикатами:

a) $2 \cdot 2 = 4$;	d) $x \cdot y = z$;	g) x і y;
b) $2 \cdot x = 4$;	e) y : x;	h) $X \cup Y$;
c) y $\cdot$ x = 4;	f) y $\vdots$ x;	i) $X \cap Y$ = Z.

2. Знайти область істинності логічної функції:

a) "місто x знаходиться на березі річки Дніпро", де $x \in$ {Львів, Канів, Вінниця, Умань, Київ};
b) "x кратне 4", де $x \in \{0, 2, 4, 6, 8, 10\}$;
c) "$\sin x \leq 1$", де $x \in R$;
d) "$\cos x > 1$", де $x \in R$;
e) "x межує з y", якщо $x, y \in$ {Португалія, Іспанія, Франція, Бельгія, Андорра, Монако};
f) "точка з координатою x лежить лівіше від точки з координатою y", де $x, y \in \{-3, -1, 0, 1, 2\}$;
g) "$x + y = z$", де $x, y, z \in \{0, 1, 2, 3, 4\}$;
h) "$\sqrt[x]{y} = z$", де $x, y, z \in \{0, 1, 2, 3, 4\}$;
i) "ріка x впадає у море y" де $x \in$ {Дніпро, Дунай, Волга, Рейн, Сена}, $y \in$ {Балтійське, Чорне, Норвезьке, Середземне}.

3.2. ФОРМУЛИ ЛОГІКИ ПРЕДИКАТІВ

Після того, як означені всі операції логіки предикатів, дамо індуктивне означення формули логіки предикатів.

ОЗНАЧЕННЯ 3.2.1. *1) $P_k(x_1, ..., x_n)$ ($k \in N$, $n \in N_0$) — формула логіки предикатів. Цю формулу називають атомарною або елементарною.*

2) Якщо A_1 і A_2 — формули логіки предикатів, то $(\rceil A_1)$, $(A_1 \wedge A_2)$, $(A_1 \vee A_2)$, $(A_1 \Rightarrow A_2)$, $(A_1 \Leftrightarrow A_2)$ — також формули.

3) Якщо A — формула логіки предикатів, x вільна змінна в A, то $\forall x(A)$, $\exists x(A)$ також формули, в яких змінна x зв'язана, а решта предметних змінних, що входять у формулу A вільно чи зв'язано лишаються в нових формулах відповідно такими ж.

4) Ніяких інших формул логіки предикатів немає.

Приклад 3.2.1. Вираз

$$(\forall x_1(\forall x_2(\exists x_3((P_1(x_1, x_2, x_3) \Rightarrow P_2(x_1, x_3)))))) \qquad (3.2.1)$$

є формулою логіки предикатів.

Дійсно,

$P_1(x_1, x_2, x_3)$, $P_2(x_1, x_3)$ є формули згідно п.1) означення;

$(P_1(x_1, x_2, x_3) \Rightarrow P_2(x_1, x_3))$ є формула згідно п. 2) означення;

$(\exists x_3((P_1(x_1, x_2, x_3) \Rightarrow P_2(x_1, x_3))))$, $(\forall x_2(\exists x_3((P_1(x_1, x_2, x_3) \Rightarrow P_2(x_1, x_3)))))$, $(\forall x_1(\forall x_2(\exists x_3((P_1(x_1, x_2, x_3) \Rightarrow P_2(x_1, x_3))))))$

це формули згідно п. 3) означення.

Приклад 3.2.2. Вираз

$$(\exists x_1(P_1(x_2, x_3) \wedge (\forall x_1(P_2(x_1))))) \qquad (3.2.2)$$

не є формулою логіки предикатів.

Дійсно,

$P_1(x_2, x_3)$, $P_2(x_1)$ — формули згідно п. 1) означення;

$\forall x_1(P_2(x_1))$ — формула згідно п. 3) означення;

$(P_1(x_2, x_3)\wedge(\forall x_1(P_2(x_1))))$ — формула згідно п. 2) означення.

Проте в останньому виразі x_1 не є вільною змінною, і п. 3) означення до нього застосувати не можна. Отже, за п.4) формула (3.2.2), не є формулою логіки предикатів.

З означення формули логіки предикатів випливає, що операція квантифікації або навішування квантора на формулу, зменшує в останній число вільних змінних за пунктом 3) означення.

ОЗНАЧЕННЯ 3.2.2. *Формула логіки предикатів, яка не містить жодної вільної предметної змінної називається замкненою формулою.*

Значення замкненої формули не залежить від жодної предметної змінної. Це висловлення істинне або хибне. Значення формули, яка містить хоча б одну предметну змінну, суттєво залежить від останньої, це висловлювальна форма або відкрита формула.

Приклади замкнених формул: P, $\exists x(P_1(x) \Rightarrow \forall y P_2(y))$, $(\forall x_1(\forall x_2(\exists x_3((P_1(x_1, x_2, x_3) \Rightarrow P_2(x_3))))))$.

Мають місце всі умови скорочення числа дужок прийняті в алгебрі висловлень. Крім того, вважаємо, що квантори зв'язують сильніше, ніж операції алгебри висловлень. Опускаємо також дужки, що визначають область дії квантора, тоді, коли остання є елементарною формулою.

Якщо квантори слідують один за одним, то опускаємо дужки між ними, виконуючи квантифікацію в порядку, зворотному до їх написання.

Наприклад, формула (2.1) після застосування правил скорочення дужок матиме вигляд:

$$\forall x_1 \forall x_2 \exists x_3(P_1(x_1, x_2, x_3) \Rightarrow P_2(x_1, x_3)).$$

Семантика формул логіки предикатів. Інтерпретація

В алгебрі висловлень для визначення істинісних значень формули складали таблиці істинності. Але в логіці предикатів формули визначені на множинах, які можуть містити багато елементів, бути нескінченними, навіть незчисленними.

У логіці висловлень оперували з пропозиційними змінними, значеннями яких були висловлення, тобто об'єкти, які можуть приймати тільки два значення: істинне або хибне. Було лише невідомо, яке саме значення має місце. Складаючи таблиці істинності, враховували всі випадки.

У формулах логіки предикатів містяться предикатні змінні, визначені на універсальній множині, значеннями яких є конкретні предикати.

Якщо в формулі логіки предикатів замість кожної предикатної змінної підставити конкретний предикат, визначений на деякій обраній множині M, то формула перетвориться в конкретний предикат, заданий на множині M. При цьому, якщо вихідна формула була замкненою, то отриманий конкретний предикат виявиться нульмістним, тобто буде висловленням. Якщо ж вихідна формула була відкритою, тобто містила вільні входження предметних змінних, то в результаті підстановки отримаємо предикат, що залежить від деяких предметних змінних. Якщо тепер підставити замість цих предметних змінних конкретні предмети з множини M, то отриманий предикат, а в остаточному підсумку — вихідна формула, перетвориться в конкретне висловлення.

Перетворення формули логіки предикатів у висловлення описаним вище методом, а також саме одержане висловлення, називається *інтерпретацією* цієї формули на множині M.

Нехай $A(P_1, ..., P_k)$ — формула логіки предикатів, яка містить n-місні предикатні змінні $P_1, ..., P_k$; D — певна структура (непорожня множина M з заданими на ній n-місними відношеннями $R_1, ..., R_k$ між її елементами).

ОЗНАЧЕННЯ 3.2.3. *Інтерпретацією формули $A(P_1, ..., P_k)$ в D називають заміну кожної n-місної предикатної змінної в A ($n \in N$) відповідним n-місним відношенням в M, та кожної предметної сталої деяким елементом M. При цьому вважаємо, що вільні предметні змінні в A пробігають M — множину інтерпретації, а символам логічних операцій алгебри висловлень і кванторам в A надається їх звичайний зміст.*

Інтерпретацією формули логіки предикатів, яка не містить жодної вільної предметної змінної (замкненої форми), є певне висловлення.

Приклад 3.2.3. Дамо інтерпретацію формулі $\forall x \exists y P(x, y)$. За множину M візьмемо множину всіх чоловіків, що живуть або жили, а замість предикатної змінної $P(x, y)$ підставимо конкретний предикат, визначений на M: "x є батьком y". Тоді вихідна формула перетвориться в таке (зрозуміло, хибне) висловлення $\forall x \exists y$ (x є батьком y) або звичайною мовою "у кожного чоловіка є син". Цій же формулі можна дати іншу інтерпретацію. Візьмемо за M множину N всіх натуральних чисел, а замість предикатної змінної $P(x, y)$ підставимо предикат "$x < y$", визначений на N. Тоді вихідна формула перетвориться у (зрозуміло, істинне) висловлення $\forall x \exists y$ $(x < y)$ — "для кожного натурального числа існує більше нього натуральне число".

Складність дослідження формул логіки предикатів швидко росте із зростанням числа елементів множини інтерпретації і числа аргументів у предикатів, наявних у формулі.

Розглянемо випадок коли множина інтерпретації M містить два елементи, а предикати, які входять у дану формулу, двомісні.

Кількість різних одномісних предикатів на множині $\{a, b\}$ дорівнює 4. Випишемо їх у таку таблицю:

x	$L_1(x)$	$L_2(x)$	$L_3(x)$	$L_4(x)$
a	0	0	1	1
b	0	1	0	1

На тій же множині кількість двомісних предикатів дорівнює 16:

(x, y)	$L_1(x, y)$	$L_2(x, y)$	$L_3(x, y)$	$L_4(x, y)$	$L_5(x, y)$	$L_6(x, y)$	$L_7(x, y)$	$L_8(x, y)$	$L_9(x, y)$	$L_{10}(x, y)$	$L_{11}(x, y)$	$L_{12}(x, y)$	$L_{13}(x, y)$	$L_{14}(x, y)$	$L_{15}(x, y)$	$L_{16}(x, y)$
(a, a)	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
(a, b)	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
(b, a)	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
(b, b)	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Приклад 3.2.4. Оцінити формулу логіки предикатів

$$A = \forall x(P(x) \Rightarrow P(y)) \vee \exists z \neg P(z) \tag{3.2.3}$$

на двоелементній множині $\{a, b\}$.

Розв'язання: Предикатна змінна P у формулі (3.2.3) пробігає множину одномісних логічних функцій L_1–L_4 на $\{a, b\}$, вільна предметна змінна y пробігає множину $\{a, b\}$. Далі застосовуємо операції алгебри висловлень, причому враховуємо, що на $\{a, b\}$ квантор загальності перетворюється в кон'юнкцію, а квантор існування — в диз'юнкцію. Результати заносимо в таблицю:

№ з/п	P	y	$\forall x(P(x) \Rightarrow P(y))$	$\neg P(z)$	A
1.	$L_1(x)$	a	1	1	1
2.	$L_1(x)$	b	1	1	1
3.	$L_2(x)$	a	0	1	1
4.	$L_2(x)$	b	1	0	1
5.	$L_3(x)$	a	1	0	1
6.	$L_3(x)$	b	0	1	1
7.	$L_4(x)$	a	1	0	1
8.	$L_4(x)$	b	1	0	1

Розглянемо детально обчислення значення формули в третьому та четвертому стовпчику таблиці.

1. P-L_1, y-a: $\forall x(L_1(x) \Rightarrow L_1(a)) = (L_1(a) \Rightarrow L_1(a)) \wedge (L_1(b) \Rightarrow L_1(a)) = (0 \Rightarrow 0) \wedge (0 \Rightarrow 0) = 1 \wedge 1 = 1.$
$\exists z(\neg L_1(z)) = (\neg L_1(a) \vee \neg L_1(b)) = 1 \vee 1 = 1.$
Враховуючи значення попередніх рядків маємо: $1 \vee 1 = 1$.
2. P-L_1, y-b: $\forall x(L_1(x) \Rightarrow L_1(b)) = (L_1(a) \Rightarrow L_1(b)) \wedge (L_1(b) \Rightarrow L_1(b)) = (0 \Rightarrow 0) \wedge (0 \Rightarrow 0) = 1 \wedge 1 = 1.$
$\exists z(\neg L_1(z)) = (\neg L_1(a) \vee \neg L_1(b)) = 1 \vee 1 = 1.$
Враховуючи значення попередніх рядків маємо: $1 \vee 1 = 1$.
3. P-L_2, y-a: $\forall x(L_2(x) \Rightarrow L_2(a)) = (L_2(a) \Rightarrow L_2(a)) \wedge (L_2(b) \Rightarrow L_2(a)) = (0 \Rightarrow 0) \wedge (1 \Rightarrow 0) = 1 \wedge 0 = 0.$
$\exists z(\neg L_2(z)) = (\neg L_2(a) \vee \neg L_2(b)) = 1 \vee 0 = 1.$
Враховуючи значення попередніх рядків маємо: $0 \vee 1 = 1$.
4. P-L_2, y-b: $\forall x(L_2(x) \Rightarrow L_2(b)) = (L_2(a) \Rightarrow L_2(b)) \wedge (L_2(b) \Rightarrow L_2(b)) = (0 \Rightarrow 1) \wedge (1 \Rightarrow 1) = 1 \wedge 1 = 1.$
$\exists z(\neg L_2(z)) = (\neg L_2(a) \vee \neg L_2(b)) = 1 \vee 0 = 1.$

Враховуючи значення попередніх рядків маємо: $1 \vee 1 = 1$.

5. P-L_3, y-a: $\forall x(L_3(x) \Rightarrow L_3(a)) = (L_3(a) \Rightarrow L_3(a)) \wedge (L_3(b) \Rightarrow L_3(a)) = (1 \Rightarrow 1) \wedge (0 \Rightarrow 1) = 1 \wedge 1 = 1$.

$$\exists z(\rceil L_3(z)) = (\rceil L_3(a) \vee \rceil L_3(b)) = 0 \vee 1 = 1.$$

Враховуючи значення попередніх рядків маємо: $1 \vee 1 = 1$.

Обчислення у рядках 6, 7, 8 виконайте самостійно.

Такий процес оцінки формул логіки предикатів виявляється дуже громіздким навіть для випадку двоелементної універсальної множини, проте він дає змогу зрозуміти саму суть механізму оцінки формул логіки предикатів.

ОЗНАЧЕННЯ 3.2.4. *Формула A логіки предикатів називається виконуваною (спростовною) на множині M, якщо існує хоча б одна інтерпретація A на M і хоча б одна заміна вільних предметних змінних в A назвами елементів M, при яких отримаємо істинне (хибне) висловлення.*

ОЗНАЧЕННЯ 3.2.5. *Формула, яка на деякій структурі D істинна при всіх інтерпретаціях і всіх замінах вільних предметних змінних назвами елементів з D, називається істинною або загальнозначущою на даній структурі.*

Якщо формула істинна на множині M, то вона істинна і на M_1 при умові, що потужність M_1 не більша потужності M. Дійсно, всі інтерпретації і заміщення вільних предметних змінних на M_1 є тоді відповідно інтерпретаціями і заміщеннями вільних предметних змінних на M. Це твердження виконується для формул, які не містять ні символів конкретних предикатів, ні символів предметних сталих. При наявності останніх дане твердження немає сили.

ОЗНАЧЕННЯ 3.2.6. *Формула A називається виконуваною, якщо існує множина на якій A — виконувана.*

Якщо формула логіки предикатів (яка не містить символів конкретних предикатів, ні символів предметної сталої) є виконуваною на множині M, то вона виконується на будь-якій множині M_1, такій, що потужність M_1 не менша від потужності M.

Зрозуміло, що для доведення виконуваності формули A на кожній непорожній множині досить довести, що формула A виконувана на одноелементній множині.

У позначеннях формул ми часто користуємося символом $A(x)$. Цей запис вживається в основному для того, щоб вказати на залежність формули $A(x)$ від x. Тоді природно щоб $A(y)$ означало б для y те саме, що $A(x)$ означає для x, або, як кажуть, щоб $A(y)$ було окремим випадком $A(x)$. Проте, якщо не зробити відповідних застережень, то такий факт справджується не завжди.

Наприклад, нехай універсальною множиною є множина всіх натуральних чисел N, а формула $A(x)$ має вигляд $\exists y$ $(x = 2y)$, тобто $A(x)$ означає, що x є парним числом. Тоді $A(y)$ матиме вигляд $\exists y(y = 2y)$. Те, що стверджується про y у формулі $A(y)$, у даному разі не є тим самим, що стверджується про x в $A(x)$. Інакше кажучи, $A(y)$ тут не можна розглядати як окремий випадок $A(x)$. Справа, очевидно, в тому, що входження y в $A(x)$ вільне, а в $A(y)$ — зв'язане. При цьому утворення $A(y)$ з $A(x)$ є, так би мовити, незаконним, і, щоб запобігти цьому, вводиться спеціальне застереження.

ОЗНАЧЕННЯ 3.2.7. *Предметна змінна y називається допустимою для підстановки в формулу A (x) замість x (або вільною для x в A(x)), якщо жодне вільне входження y в A(x) не стає зв'язаним в наслідок цієї підстановки.*

Необхідною і достатньою умовою правильності, "законності" утворення $A(y)$ із $A(x)$ є допустимість y для підстановки в формулу $A(x)$ замість x.

Наприклад:

$$A(x) = \exists z\ (x = z + 1);\ A(y) = \exists z\ (y = z + 1).$$

Тут y допустима для підстановки в формулу $A(x)$ замість x. Утворення $A(y)$ із $A(x)$ — правильне.

Вправи

1. Які з наступних записів є формулами :

а) $\forall P(x)$;
б) $\forall x P(x)$;
в) $\exists x, \exists y P(x, y)$;
г) $(\forall x(\forall y P(x, y)))$;
д) $\exists x, y P(x, y)$;
е) $\forall xy P(x, y)$;
є) $(\exists x) P(x)$;
ж) $\forall x P(x) \vee \exists y Q(x, y)$.

2. Опустити дужки у формулах:

а) $(\forall x(\exists y(\forall z P(x, y, z))))$;
б) $((\forall x P(x)) \Rightarrow \mathrm{P}(x, y))$;
в) $(\forall x(\mathrm{P}(x) \vee \mathrm{P}(x, y)))$;
г) $((\exists x \mathrm{P}(x)) \Rightarrow ((\forall x \mathrm{Q}(x)) \vee \rceil \mathrm{Q}(x)))$.

3. Поновити дужки у формулах:

а) $\forall x P(x) \vee Q(x) \Rightarrow \exists x Q(x) \wedge P(x)$;
б) $P(x) \Rightarrow \forall x Q(x) \vee \rceil Q(x)$;
в) $\forall x P(x,y) \Rightarrow \forall y Q(y)$;
г) $\forall x \forall y\ \forall z\ P(x, y, z) \Rightarrow Q(y) \vee \rceil P(x)$;
д) $\exists x \forall y\ \exists z\ P(x, z) \wedge \exists y \rceil \forall z P(y, z)$;
е) $\rceil \forall z\ P(x, z) \Rightarrow \exists y Q(y) \Rightarrow P(x, y) \wedge P(y)$.

3.3. ЛОГІЧНОЗАГАЛЬНОЗНАЧУЩІ ФОРМУЛИ ЛОГІКИ ПРЕДИКАТІВ

ОЗНАЧЕННЯ 3.3.1. *Формула логіки предикатів, яка є істинною на кожній множині, при кожній інтерпретації, називається тотожно істинною або логічно загальнозначущою.*

Формула A є логічно загальнозначущою (лзз) тоді і тільки тоді, коли при всіх інтерпретаціях на яких завгодно множинах і при всіх заміщеннях вільних предметних змінних, що входять в A, назвами елементів з цих множин, вона перетворюється в істинне висловлення. Логічно загальнозначущі формули є схемами логічно-істинних тверджень, законами логіки. Якщо формула A є логічно загальнозначущою, то позначатимемо $\models A$.

Як вже зазначалось, мова логіки предикатів більш точніша ніж мова алгебри висловлень, і тому тавтології логіки предикатів (лзз) точніше відображають процеси логічних умовиводів.

Розгляд тавтологій логіки предикатів почнемо з встановлення того, що найпростіші з них отримуються з тавтологій алгебри висловлень, а тавтології алгебри висловлень є частиною тавтологій логіки предикатів.

ТЕОРЕМА 3.3.1. *Кожна формула, яка отримується з тавтології алгебри висловлень заміною пропозиційних змінних, які до неї входять, довільними предикатними змінними, є логічно загальнозначуща.*

Доведення. Нехай $A(a_1, a_2, ..., a_n)$ — тавтологія алгебри висловлень і $P_1(x_1, ..., x_m)$, $P_2(y_1, ..., y_m)$, ..., $P_n(z_1, ... , z_m)$ — предикатні змінні. Підставимо їх у дану формулу замість пропозиційних змінних $a_1, a_2, ..., a_n$ відповідно. Отримаємо формулу логіки предикатів:

$$A(P_1(x_1, ..., x_m), P_2(y_1, ..., y_m), ... , P_n(z_1, ..., z_m)).$$

Якщо тепер замість предикатних змінних підставити довільні конкретні предикати $B_1(x_1, ..., x_m), ... , B_n(z_1, ..., z_m)$, то формула перетвориться у конкретний предикат $A(B_1(x_1, ..., x_m), B_2(y_1, ..., y_m), ..., B_n(z_1, ..., z_m))$. Цей предикат тотожно істинний, тому що підстановка замість предметних змінних $x_1, ..., x_m, ..., z_1, ..., z_m$ назв довільних конкретних предметів $b_1, ..., b_m, ..., d_1, ..., d_m$ перетворює даний предикат у висловлення $A(B_1(b_1, ..., b_m), ..., B_n(d_1, ..., d_m))$, яке можна отримати також в результаті підстановки у вихідну тавтологію $A(a_1, a_2, ..., a_n)$ алгебри висловлень замість пропозиційних змінних $a_1, a_2, ..., a_n$ конкретних висловлень $B_1(b_1, ..., b_m), ..., B_n(d_1, ..., d_m)$ відповідно, і тому є істинним. Тоді формула $A(P_1, P_2, ..., P_n)$ логіки предикатів також є істинною, причому на кожній множині. Отже, за означенням, ця формула є логічно загальнозначуща. Теорему доведено.

Наведемо список деяких тавтологій логіки предикатів:

1. $\forall x\, P(x) \Rightarrow P(y)$;
2. $P(y) \Rightarrow \exists x\, P(x)$;
3. $\rceil \forall x P(x) \Leftrightarrow \exists x\, \rceil P(x)$
4. $\rceil \exists x\, P(x) \Leftrightarrow \forall x \rceil (P(x)$

$\Big\}$ Закони де Моргана для кванторів;

5. $\forall x(P(x) \wedge Q(x)) \Leftrightarrow \forall x P(x) \wedge \forall x Q(x)$;
6. $\exists x(P(x) \vee Q(x)) \Leftrightarrow \exists x\, P(x) \vee \exists x\, Q(x)$;
7. $\forall x\, P(x) \vee \forall x\, Q(x) \Rightarrow \forall x(P(x) \vee Q(x))$;
8. $\exists x(P(x) \wedge Q(x)) \Rightarrow \exists x P(x) \wedge \exists x Q(x)$;
9. $\forall x(A(x) \wedge B) \Leftrightarrow \forall x A(x) \wedge B$;
10. $\forall x(A(x) \vee B) \Leftrightarrow \forall x A(x) \vee B$;
11. $\exists x(A(x) \wedge B) \Leftrightarrow \exists x A(x) \wedge B$;
12. $\exists x(A(x) \vee B) \Leftrightarrow \exists x A(x) \vee B$;
13. $\forall x(A(x) \Rightarrow B) \Leftrightarrow \exists x\, A(x) \Rightarrow B$;
14. $\exists x(A(x) \Rightarrow B) \Leftrightarrow \forall x\, A(x) \Rightarrow B$;
15. $\forall x(B \Rightarrow A(x)) \Leftrightarrow B \Rightarrow \forall x A(x)$;
16. $\exists x(B \Rightarrow A(x)) \Leftrightarrow B \Rightarrow \exists x A(x)$;
17. $\forall x\, \forall y\, A(x, y) \Leftrightarrow \forall y\, \forall x A(x, y)$;
18. $\exists x\, \exists y\, A(x, y) \Leftrightarrow \exists y\, \exists x\, A(x, y)$;
19. $\exists x\, \forall y\, A(x, y) \Rightarrow \forall y\, \exists x\, A(x, y)$.

Приклад 3.3.1. Доведемо, що формула 3 є логічно загальнозначуща. Дана формула замкнена, тобто не має вільних предметних змінних. Тому підставивши в цю формулу замість предикатної змінної *P(x)* довільний конкретний одномісний предикат $A(x)$, визначений на деякій множині *M*, одержимо висловлення

$$\rceil \forall x A(x) \Leftrightarrow \exists x\, \rceil A(x) \qquad (*)$$

Для доведення його істинності потрібно переконатися, що обидві частини еквівалентності одночасно істинні або одночасно хибні. Справді, висловлення $\rceil \forall x A(x)$ істинне тоді і тільки тоді, коли висловлення $\forall x A(x)$ хибне, що можливо, на підставі означення, тоді і тільки тоді, коли предикат *A(x)* спростовний. Далі, спростовність предиката *A(x)* означає виконуваність його заперечення $\rceil A(x)$ (обміркуйте це!), що рівносильне, на підставі означення, істинності висловлення $\exists x \rceil A(x)$. Отже, висловлення $\rceil \forall x A(x)$ істинне тоді і тільки тоді, коли істинне висловлення $\exists x \rceil A(x)$. Тоді, висловлення (*) істинне, що і доводить, що формула 3 є логічно загальнозначуща.

Приклад 3.3.2. Доведемо, що формула 5 є логічно загальнозначуща.

Для доведення еквіваленції доведемо дві імплікації:

$$\forall x(P(x) \wedge Q(x)) \Rightarrow \forall x\, P(x) \wedge \forall (x\, Q(x), \qquad (3.3.1)$$

$$\forall x\, P(x) \wedge \forall x\, Q(x) \Rightarrow \forall x\, (P(x) \wedge Q(x)). \qquad (3.3.2)$$

Припустимо, що (3.3.1) не є логічно загальнозначуща, тобто знайдеться множина M та інтерпретація P^* і Q^* замість P і Q, що утворене в наслідок інтерпретації висловлення хибне. За означенням імплікації маємо

$$|\forall x(P^*(x) \wedge Q^*(x))| = 1, \qquad (3.3.3)$$

$$|\forall x\, P^*(x) \wedge \forall x\, Q^*(x)| = 0. \qquad (3.3.4)$$

Із співвідношення (3.3.3) для кожного $a \in M$ матимемо

$$|P^*(a) \wedge Q^*(a)| = 1,$$

а це означає, що $|P^*(a)| = 1$ і $|Q^*(a)| = 1$ для кожного елемента a з множини M.

Із співвідношення (3.3.4) випливає, що знайдеться такій $b \in M$ або $c \in M$, що $|P^*(b)| = 0$ або $|Q^*(c)| = 0$. А це суперечить попередньому реченню. Суперечність свідчить про те, що зроблене припущення неправильне, тобто формула (3.3.1) є логічно загальнозначущою.

Аналогічно можна довести що формула (3.3.2) — тотожно істинна.

Отже, формула 5 є логічно загальнозначуща.

Приклад 3.3.3. Доведемо, що формула 13 логічно загальнозначуща.

Зауважимо, що предикатна змінна B в цій формулі може бути не тільки нульмісною, але і будь-якою n-місною, важливо лише, щоб у неї не входила предметна змінна x. Отже, нехай B є $B(y_1, ..., y_n)$. Будемо вважати для стислості, що B є одномісна предикатна змінна $B(y)$.

Припустимо, що дана формула не є тавтологією. Тоді існують такі конкретні предикати A^* і $B^*(y)$, визначені на множині M, що предикат (від y) $\forall x(A^*(x) \Rightarrow B^*(y)) \Leftrightarrow \exists x\, A^*(x) \Rightarrow B^*(y)$ перетворюється в помилкове висловлення при підстановці замість предметної змінної у деякого конкретного предмета $b \in M$:

$$|\forall x(A^*(x) \Rightarrow B^*(b)) \Leftrightarrow \exists x\, A^*(x) \Rightarrow B^*(b)| = 0.$$

Еквівалентність хибна, якщо її члени приймають різні значення істинності, тобто тут можуть трапитись два випадки. Перший

$$|\forall x(A^*(x) \Rightarrow B^*(b))| = 1, \qquad (3.3.5)$$

$$|\exists x\, A^*(x) \Rightarrow B^*(b)| = 0; \qquad (3.3.6)$$

і другий

$$|\forall x(A^*(x) \Rightarrow B^*(b))| = 0, \qquad (3.3.7)$$

$$|\exists x\, A^*(x) \Rightarrow B^*(b)| = 1; \qquad (3.3.8)$$

Розглянемо першу можливість. З (3.3.6) за означенням імплікації:

$$|\exists x(A^*(x))| = 1, \qquad (3.3.9)$$

$$|B^*(b)| = 0. \qquad (3.3.10)$$

Далі, із (3.3.9) за означенням квантора існування випливає, що предикат $A(x)$ виконуваний, тобто

$$|A^*(a)| = 1, \tag{3.3.11}$$

для деякого $a \in M$. Повернемося до співвідношення (3.3.5). За означенням квантора загальності предикат $A^*(x) \Rightarrow B^*(b)$ тотожно істинний. Зокрема, якщо замість предметної змінної x підставити $a \in M$, то отримаємо істинне висловлення $|A^*(a) \Rightarrow B^*(b)| = 1$.

Але, з огляду на (3.3.10) і (3.3.11), отримаємо
$|A^*(a) \Rightarrow B^*(b)| = |A^*(a)| \Rightarrow |B^*(b)| = 1 \Rightarrow 0 = 0$ – суперечність.

Розглянемо другу можливість, виражену в співвідношеннях (3.3.7), (3.3.8). З (3.3.7), за означенням квантора загальності, випливає, що $|A^*(a) \Rightarrow B^*(b)| = 0$, для деякого $a \in M$. Тоді, за означенням імплікації:

$$|(A^*(a)| = 1, \ | B^*(b))| = 0. \tag{3.3.12}$$

Через останнє співвідношення, із співвідношення (3.3.8) випливає, що $|\exists x A^*(x)| = 0$. Останнє означає тотожну хибність предиката $A(x)$. Зокрема, для предмета $a \in M$ маємо $A(a) = 0$, що суперечить першому із співвідношень (3.3.12).

Отже, у кожному випадку приходимо до суперечності, що доводить неможливість зробленого припущення. Тоді, формула 13 — тавтологія.

Зауважимо, що закони 1–8 сформульовані для атомарних предикатів P і Q, а не для довільних формул логіки предикатів A, B. Це має істотне значення тільки для законів 1, 2, оскільки в останніх поряд з $P(x)$ фігурує $P(y)$. Як зазначалось, такий перехід не завжди правомірний. Тому закони 1, 2 мають місце для довільної формули логіки предикатів $A(x)$ замість $P(x)$ тільки при умові, що змінна y — вільна для x у формулі $A(x)$. У іншому випадку формули 1, 2 перестають бути логічно загальнозначущими.

Проведені доведення не дають відповіді на питання, чи є довільно задана формула логіки предикатів логічно загальнозначуща.

ТЕОРЕМА 3.3.2. *A — логічно загальнозначуща формула логіки предикатів тоді й тільки тоді, коли $\rceil A$ — невиконувана формула.*

Доведення. Нехай M — довільна множина, A^* — результат довільної інтерпретації і заміщення вільних предметних змінних формули A. Дано $|A^*|=1$. Тоді $|\rceil A^*|=0$, тобто $\rceil A$ — невиконувана. Навпаки нехай $\rceil A$ — невиконувана. Тоді на кожній множині, при кожній інтерпретації $|\rceil A|=0$, тобто на кожній множині, при кожній інтерпретації $|A^*|=1$. За означенням A — логічно загальнозначуща формула логіки предикатів.

ОЗНАЧЕННЯ 3.3.2. *Формули A та B логіки предикатів називаються еквівалентними на множині M, якщо при всіх інтерпретаціях на M і при всіх заміщеннях вільних предметних змінних назвами елементів M формули A та B набувають однакові значення істинності.*

ОЗНАЧЕННЯ 3.3.3. *Формули A та B еквівалентні на кожній множині називаються логічно еквівалентними або рівносильними.*

Рівносильність формул A та B позначатимемо $A \equiv B$.

Властивості відношення рівносильності: рефлексивність, симетричність, транзитивність.

Неважко зрозуміти на підставі означень 3.3.3 і 3.3.1, що формули A і B рівносильні тоді і тільки, коли формула $A \Leftrightarrow B$ є тавтологією логіки предикатів: $A \equiv B$ тоді і тільки тоді, коли $\models A \Leftrightarrow B$.

З тотожно істинних формул отримаємо рівносильності:

$$\left.\begin{array}{l} 1.\ \neg\forall x P(x) \equiv \exists x\, \neg P(x) \\ 2.\ \neg\exists x\, P(x) \equiv \forall x \neg(P(x) \end{array}\right\} \text{Закони де Моргана для кванторів;}$$

3. $\forall x(P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$;
4. $\exists x(P(x) \vee Q(x)) \equiv \exists x\, P(x) \vee \exists x Q(x)$;
5. $\forall x(A(x) \wedge B) \equiv \forall x A(x) \wedge B$;
6. $\forall x(A(x) \vee B) \equiv \forall x A(x) \vee B$;
7. $\exists x(A(x) \wedge B) \equiv \exists x A(x) \wedge B$;
8. $\exists x(A(x) \vee B) \equiv \exists x A(x) \vee B$;
9. $\forall x(A(x) \Rightarrow B) \equiv \exists x\, A(x) \Rightarrow B$;
10. $\exists x(A(x) \Rightarrow B) \equiv \forall x\, A(x) \Rightarrow B$;
11. $\forall x(B \Rightarrow A(x)) \equiv B \Rightarrow \forall x A(x)$;
12. $\exists x(B \Rightarrow A(x)) \equiv B \Rightarrow \exists x A(x)$;
13. $\forall x\, \forall y\, A(x, y) \equiv \forall y\, \forall x A(x, y)$;
14. $\exists x\, \exists y\, A(x, y) \equiv \exists y\, \exists x\, A(x, y)$;
15. $QxA(x) \equiv QyA(y)$;

де Q — знак $\forall$ або $\exists$, змінна y не входить в $A(x)$ і змінна x не входить в $A(y)$.

Як і в алгебрі висловлень, можна заміняти одну рівносильну формулу іншою. Перехід від однієї рівносильної формули до іншої називається *рівносильним перетворенням вихідної формули*. У процесі рівносильних перетворень формул логіки предикатів можуть використовуватися рівносильності, відомі з алгебри висловлень.

Вправи

1. Довести рівносильності:

a) $\forall x(A(x) \Rightarrow B) \equiv \exists x A(x) \Rightarrow B$;
b) $\exists x(A(x) \Rightarrow B) \equiv \forall x A(x) \Rightarrow B$;
c) $\forall x(B \Rightarrow A(x)) \equiv B \Rightarrow \forall x A(x)$;
d) $\exists x(B \Rightarrow A(x)) \equiv B \Rightarrow \exists x A(x)$.

2. Довести загальнозначущість формул:

a) $\forall x \forall y(A(x, y) \Rightarrow \forall x A(x, x)$;
b) $\exists x A(x, x)) \Rightarrow \exists x \exists y A(x, y)$;
c) $\forall x A(x) \Rightarrow \exists x B(x) \Leftrightarrow \exists x(A(x) \Rightarrow B(x))$;
d) $(\exists x A(x) \Rightarrow \forall x B(x)) \Rightarrow \forall x(A(x) \Rightarrow B(x))$;
e) $\exists x(A(y) \Rightarrow A(x))$;
f) $\neg \exists x A(x) \Rightarrow \neg \forall x A(x)$;
g) $\exists x \forall y A(x, y) \Rightarrow \forall y \exists x A(x, y)$;
h) $\exists x(A(x) \wedge B(x)) \Rightarrow \exists x A(x) \wedge \exists x B(x)$;
i) $\exists x(A(x) \vee B(x)) \Rightarrow \exists x A(x) \vee \exists x B(x)$;
j) $\forall x(A(x) \wedge B(x)) \Rightarrow \forall x A(x) \wedge \forall x B(x)$;
k) $\forall x A(x) \vee \forall x B(x) \Rightarrow \forall x(A(x) \vee B(x))$;
l) $\forall x(A(x) \Rightarrow \neg B(x)) \Rightarrow \neg(\exists x A(x) \wedge \forall x B(x))$;
m) $\exists x(A(x) \Rightarrow B(x)) \Rightarrow (\forall x A(x) \Rightarrow \exists x B(x))$.

3. З'ясувати, чи є загальнозначущими такі формули:

a) $\exists x A(x) \Rightarrow A(x)$;
b) $B(x) \Rightarrow \exists x B(x)$;
c) $\exists x \exists y(\neg A(x) \wedge \neg B(y) \vee A(x) \vee B(y)$;
d) $\exists x \forall y(A(x) \Rightarrow B(x, y))$;
e) $\forall x(A(x) \Rightarrow B(x)) \Rightarrow (\forall x A(x) \Rightarrow \forall x B(x))$;
f) $\exists x(A(x) \Rightarrow B(x)) \Rightarrow (\exists x A(x) \Rightarrow \exists x B(x))$;
g) $\forall x(A(x) \Rightarrow B(x)) \Rightarrow (\exists x A(x) \Rightarrow \exists x B(x))$.

4. З'ясувати, чи є виконуваними такі формули:

a) $\exists x A(x) \Rightarrow A(y)$;
b) $\exists x \forall y(A(x,x) \wedge \neg A(x, y))$;
c) $\exists x \exists y(B(x) \wedge \neg B(y))$;
d) $\forall x \exists y(B(x) \wedge \neg B(y))$;
e) $\forall x B(x) \wedge \neg B(y)$;
f) $\forall x B(x) \wedge \neg A(y)$.

5. Довести, що такі формули не є загальнозначущі:

a) $\forall y \exists x(A(x,y) \Rightarrow \exists x \forall y A(x,y)$;
b) $\forall x(A(y) \Rightarrow A(x))$;
c) $\exists x \exists y(B(x,y) \Rightarrow \exists x B(x,y))$;
d) $\exists x A(x) \exists x(B(x) \Rightarrow \exists x(A(x) \wedge B(x))$;
e) $\exists x B(x) \wedge \neg A(y)$;
f) $\exists x A(x, y) \Rightarrow \neg x A(x, y)$.

3.4. ПРЕНЕКСНА НОРМАЛЬНА ФОРМА. ЗАКОН ДВОЇСТОСТІ

Рівносильні перетворення дозволяють приводити формули до того чи іншого вигляду. Одне з таких представлень має назву *зведеної форми.*

ОЗНАЧЕННЯ 3.4.1. *Формула А логіки предикатів називається зведеною, якщо в А немає інших символів операцій алгебри висловлень крім* $\rceil$, $\wedge$, $\vee$, *причому знаки заперечення відносяться тільки до елементарних формул.*

ОЗНАЧЕННЯ 3.4.2. *Формула логіки предикатів, яка не містить кванторів називається відкритою.*

ОЗНАЧЕННЯ 3.4.3. *Зведена формула А знаходиться у випередженій (пренексній) нормальній формі, якщо А — відкрита або має вигляд*

$$Qx_1, Qx_2, ..., Qx_n(B), \tag{3.4.1}$$

де Q означає $\forall$ *чи* $\exists$, *B — відкрита формула, що знаходиться в області дії кванторів* Qx_k *і всі* x_k *відмінні один від одного (*$k = 1, 2, ..., n$*).*

Вираз $Qx_1Qx_2...Qx_n$ *у формулі (3.4.1) називається приставкою, а формула B — матрицею або ядром формули А.*

ОЗНАЧЕННЯ 3.4.4. *Формула, яка знаходиться у пренексній формі і рівносильна формулі А, називається пренексною формою А.*

ТЕОРЕМА 3.4.1. *Для кожної формули логіки предикатів існує пренексна форма.*

Без втрати загальності вважаємо, що формула зведена.

Доведення проведемо методом математичної індукції по числу m логічних операцій у формулі A (включаючи квантори загальності та існування).

1. $m = 0$ формула A – елементарна, тобто вона знаходиться у випередженій нормальній формі (внф).

2. Припустимо, що твердження має місце для кожної формули логіки предикатів з числом логічних операцій, меншим ніж m.

3. Доведемо, що для формули A з числом логічних операцій рівним m теорема має місце. Формула A має вигляд: а) $\rceil B_1$; б) $B_1 \wedge B_2$; в) $B_1 \vee B_2$; г) QB_1 причому за п.2 формули B_1, B_2 знаходяться у внф.

а) $A = \rceil B_1$, де B_1 має вигляд $Qx_1...Qx_n(B_o)$. Тоді, за законами де Моргана $\rceil B_1 = Q'x_1...Q'x_n(\rceil B_o)$, де Q' — квантор двоїстий квантору Q. Таким чином для $\rceil B_1$, тобто для A, існує внф.

б) $A = B_1 \wedge B_2$ де B_1 і B_2 мають вигляд $Q_1x_1...Q_1x_n(B'_1)$ і $Q_2x_1...Q_2x_n(B'_2)$ відповідно.

Щоб застосувати до формули A закони пронесення кванторів, зробимо, в разі необхідності, перейменування зв'язаних змінних так, щоб в обох приставках не було однакових змінних і щоб змінні обох приставок були відмінні від усіх вільних змінних у B'_1 і B'_2. Далі послідовно застосовуємо рівносильності 5 і 7, поки не дістанемо формули у пренексній формі.

в) Аналогічно пункту б), але застосовуємо рівносильності 6 і 8 поки не дістанемо формули у внф.

г) $A = Q(B_1)$. Тут остання операція — квантифікування.

За принципом математичної індукції доводжуване твердження має силу для формули, що містить довільне число символів логічних операцій, тобто для довільної формули логіки предикатів.

ОЗНАЧЕННЯ 3.4.5. *Зведена формула A^* називається двоїстою A, якщо A^* одержана з A заміною кожного символу $\wedge$, $\vee$, $\forall$, $\exists$ на $\vee$, $\wedge$, $\exists$, $\forall$ відповідно.*

За означенням $(A^*)^* = A$.

1. Якщо $A(P_1, ..., P_n) \equiv B(P_1, ..., P_n)$, то $A^*(P_1, ..., P_n) \equiv B^*(P_1, ..., P_n)$ — перший закон двоїстості.
2. Якщо $\models A(P_1, ..., P_n) \Rightarrow B(P_1, ..., P_n)$, то $\models B^*(P_1, ..., P_n) \Rightarrow A^*(P_1, ..., P_n)$ — другий закон двоїстості.
3. Якщо $A(P_1, ..., P_n)$ — тотожно істина формула, то $A^*(P_1, ..., P_n)$ — невиконувана формула.

Вправи

Для вказаних формул алгебри предикатів знайти рівносильну їм зведену форму:

a) $\exists x(\mathrm{P}(x) \Rightarrow \forall y \mathrm{Q}(y))$;
b) $\rceil(\forall x(\mathrm{P}(x) \Rightarrow \exists y \mathrm{Q}(y)))$;
c) $\forall x \mathrm{P}(x) \Rightarrow \rceil(\mathrm{Q}(y) \Rightarrow \forall z \mathrm{R}(z))$;
d) $(\exists x \mathrm{P}(x) \Rightarrow \forall y \mathrm{Q}(y)) \Rightarrow \mathrm{R}(z)$;
e) $\rceil(\forall x \mathrm{P}(x) \Rightarrow \mathrm{Q}(y)) \wedge \exists y(\rceil \mathrm{R}(y) \wedge \mathrm{S}(z)))$;
f) $\rceil(\forall x \exists y(\mathrm{P}(x) \Rightarrow \mathrm{P}(y)) \wedge (\mathrm{P}(y) \Rightarrow \mathrm{P}(x)))$.

3.5. ПРОБЛЕМА РОЗВ'ЯЗНОСТІ В ЛОГІЦІ ПРЕДИКАТІВ

У алгебрі висловлень було встановлено, що існує алгоритм, який дозволяє для кожної формули алгебри висловлень відповісти на питання, чи буде дана формула виконувана, тотожно істинна, чи тотожно хибна. Аналогічна проблема виникає і для формул логіки предикатів: чи існує алгоритм, який дозволяє для кожної формули логіки предикатів відповісти на питання, чи буде дана формула виконуваною, чи загальнозначущою?

Проблема розв'язності для загальнозначущості (виконуваності) полягає в тому, щоб знайти ефективний метод, який дасть змогу для будь-якої формули A логіки предикатів визначити за скінчене число кроків чи є A логічно загальнозначущою (виконуваною) чи ні.

Цю задачу Д. Гільберт назвав центральною проблемою математичної логіки. Позитивне розв'язання даної проблеми рівнозначне оволодінню конструктивним методом перевірки правильності міркувань, здатності автоматично перевірити в кожному випадку, чи випливає певне твердження з даних посилок чи ні.

ТЕОРЕМА 3.5.1. *Обидві форми проблеми розв'язності еквівалентні, тобто позитивне розв'язання проблеми загальнозначущості веде за собою її розв'язання для виконуваності, і навпаки.*

Доведення.

1) Нехай проблема розв'язності для загальнозначущості розв'язана позитивно і A — довільна формула логіки предикатів. Тоді існує алгоритм, який дозволяє за скінчене число кроків визначити чи є формула $\rceil A$ логічно загальнозначуща чи ні. Якщо $\rceil A$ — логічно загальнозначуща, то A — невиконувана, якщо $\rceil A$ не є логічно загальнозначуща, то A виконувана. Враховуючи довільність формули A, маємо позитивне розв'язання проблеми розв'язності для виконуваності.

2) Нехай проблема розв'язності для виконуваності розв'язана позитивно, A довільна формула. Існує алгоритм, який дозволяє за скінчене число кроків визначити, чи є $\rceil A$ виконуваною чи ні. Якщо $\rceil A$ — виконувана, то A не є логічно загальнозначуща, а якщо $\rceil A$ — не виконувана, то A є логічно загальнозначуща.

У 1936 році американський математик Алонзо Черч показав, що загального алгоритму не існує, тобто проблема розв'язності в логіці предикатів — нерозв'язна. Але в окремих випадках допустиме її розв'язання.

Якщо формула логіки предикатів розглядається на скінченій множині, то замість її предикатних змінних можна підставляти конкретні предикати, що визначені на цій скінченній множині. Зважаючи на те, що операції квантифікації на скінченій множині зводяться до кон'юнкції та диз'юнкції (див. 3.1), задача про виконуваність і загальнозначущість формули логіки предикатів на скінченій множині зводиться до задачі про виконуваність або загальнозначущість деякої формули алгебри висловлень. Остання задача, як ми знаємо, ефективно розв'язна. Отже для випадку, коли універсальна множина складається зі скінченного числа елементів, алгоритм розв'язання питання розв'язності для довільної формули логіки предикатів існує завжди.

ТЕОРЕМА 3.5.2. *Якщо для даної формули логіки предикатів А можна довести, що вона виконувана, (загальнозначуща) тоді і тільки тоді, коли має місце її виконуваність, (загальнозначущість) на множині з певним скінченим числом елементів k, то питання про виконуваність (загальнозначущість) формули А розв'язне для будь-якої універсальної множини.*

Доведення подамо для випадку виконуваності. Спочатку перевіримо виконуваність формули A на множині M_k, що містить рівно k елементів. Це завжди можна зробити на підставі попереднього результату. Якщо виявиться, що A — виконувана на M_k, то вона й поготів виконувана на всякій множині з більшим числом елементів, ніж k. Залишається перевірити виконуваність A на множинах з числом елементів 1, 2, ..., $k-1$, що завжди можливо за попереднім результатом.

Виникає питання: чи не можна, виходячи з розглянутих тверджень, досягти повного розв'язання проблеми вирішення в логіці предикатів, хоча б для виконуваності.

Ні не можна. Про це свідчить факт існування таких формул логіки предикатів, які є невиконуваними на кожній скінченій множині, але в той самий час вони виконувані на нескінченній множині.

Наведемо приклад такої формули:

$$\forall x \exists y P(x, y) \wedge \forall x \overline{P}(x, x) \wedge \forall x \forall y \forall z (P(x, y) \wedge P(y, z) \Rightarrow P(x, z)).$$

Можна сказати, що вона характеризує нерефлексивність (другий член кон'юнкції) і транзитивність (третій член кон'юнкції) деякого двомісного предиката $P(x, y)$. Ця замкнена формула перетворюється в істинне висловлення якщо в неї замість предикатної змінної $P(x, y)$ підставити двомісний предикат "$x < y$", визначений на множині всіх натуральних чисел:

$$\forall x \exists y (x < y) \wedge \forall x \overline{(x < x)} \wedge \forall x \forall y \forall z ((x < y) \wedge (y < z) \Rightarrow (x < z)).$$

Покажемо, що ця формула не виконувана на жодній скінченній множині. Припустимо супротивне, тобто, що існує конкретний предикат $A(x, y)$, визначений на скінченній множині M, такий, що висловлення

$$\forall x \exists y A(x,y) \wedge \forall x \overline{A}(x,x) \wedge \forall x \forall y \forall z (A(x,y) \wedge A(y,z) \Rightarrow A(x,z)) \quad (3.5.1)$$

істинне. Тоді істинним буде і кожний член кон'юнкції (3.5.1). Зокрема, істинне висловлення $\forall x \exists y A(x, y)$. Візьмемо елемент $b_1 \in M$. Тоді з істинності останнього висловлення випливає: існує такий елемент $b_2 \in M$, що висловлення $A(b_1, b_2)$ істинне. Далі, аналогічно існує такий елемент $b_3 \in M$, що істинне висловлення $A(b_2, b_3)$, і так далі. Оскільки множина M скінченна, то не всі елементи $b_1, b_2, b_3, \ldots$ попарно різні. Нехай $b_p = b_{p+q}$ $(q>0)$. Тоді із істинності висловлень $A(b_p, b_{p+1})$, $A(b_{p+1}, b_{p+2})$, ..., $A(b_{p+q-1}, b_{p+q})$ в силу істинності висловлення (третій член кон'юнкції (5.1)) $\forall x \forall y \forall z(A(x, y) \wedge A(y, z) \Rightarrow A(x, z))$ робимо висновок, що істинне висловлення $A(b_p, b_{p+q})$, тобто висловлення $A(b_p, b_p)$. Але це суперечить істинності висловлення (другий член кон'юнкції (3.5.1)): $\forall x \rceil A(x, x)$. Отримана суперечність доводить, що жоден конкретний предикат, визначений на скінченій множині, не може перетворити дану формулу в істинне висловлення, тобто дана формула не виконувана на кожній скінченій множині.

ТЕОРЕМА 3.5.3. *Нехай А формула логіки предикатів, яка містить тільки k одномісних предикатних змінних і не містить жодних інших предикатних змінних. Для того щоб формула А була логічно загальнозначуща, необхідно і достатньо, щоб А була загальнозначущою на множині, яка містить точно 2^k елементів.*

Необхідність доводиться тривіально. За означенням, якщо A є логічно загальнозначуща, то A загальнозначуща на кожній множині, зокрема на такій, що містить точно 2^k елементів.

Достатність. Вважаємо, що A замкнена формула. Дійсно, якщо $x_1, \ldots, x_n$ — всі вільні предметні змінні в A, то формула $\forall x_1, \ldots, \forall x_n A$ одночасно з A є логічно загальнозначуща і замкнена. Вважаємо також, що A знаходиться у пренексній формі. Предикатні змінні в A позначимо $P_1, \ldots, P_k$.

Дано, що формула A — загальнозначуща на кожній множині, яка має точно 2^k елементів, тобто і на кожній множині, з числом елементів меншим, ніж 2^k. Необхідно довести, що $\models A$.

Припустимо супротивне, тобто, що існує множина M_1, така, що містить більше ніж 2^k елементів і на ній конкретні предикати $P^*_1, \ldots, P^*_k$, такі що $|A(P^*_1, \ldots, P^*_k)| = 0$.

З цього припущення виведемо хибне висловлення на множині, яка складається не більш ніж із 2^k елементів, що буде суперечити умові теореми. Для цього розіб'ємо елементи множини M_1 на класи, відносячи $a \in M_1$ і $b \in M_1$ до одного класу, тоді і тільки тоді, коли виконуються співвідношення $P_1(a) \leftrightarrow P_1(b), \ldots, P_i(a) \leftrightarrow P_i(b), \ldots, P_k(a) \leftrightarrow P_k(b)$.

При такому розбитті два елементи M_1, з одного класу не розрізняються між собою відносно кожного із предикатів P_i.

Внаслідок розбиття маємо n класів, причому $n \leq 2^k$. Дійсно, $|P_1^*(a)| = 1$ або $|P_1^*(a)| = 0$. Аналогічно $P_2^*(a)$, ..., $P_k^*(a)$. Тобто буде 2^k всіх розподілів істинісних значень, але деякі класи можуть бути порожніми, тому $n \leq 2^k$.

Позначимо класи розбиття через α_1, α_2,..., α_n $(n \leq 2^k)$. Приймемо ці класи за нову систему предметних змінних і введемо предикати Q_1, ..., Q_j, ..., Q_k, вважаючи що Q_i виконується для класу α_r тоді і тільки тоді, коли $|P^*_i(e_r)| = 1$, де e_r довільний елемент класу α_r.

Якщо тепер у висловленні A в якому є предикати P^*_i (i=1, ..., k) замінимо кожне P^*_i на Q_j, а кожну предметну змінну класом, якому вона належить, то A перейде в еквівалентне висловлення. Істинність цього твердження випливає із самої суті побудови класів α_r. Доведемо це за індукцією, вважаючи, що A знаходиться у пренексній нормальній формі. Якщо число кванторів в A рівне 0, то істинність твердження очевидна за означенням предикатів Q_j. Припустимо, що це твердження має місце при довжині приставки m. Від навішування квантора, тобто при переході від довжини приставки m до m+1 значення істинності твердження не зміниться. Тобто, це твердження справедливе для довільної довжини приставки. Зокрема, хибне на множині M_1 висловлення $A(P^*_1, ..., P^*_k)$ переходить у хибне висловлення і віднесене до множини класів α_r, яка містить n елементів $(n \leq 2^k)$. Але це суперечить загальнозначущості A на множині, яка містить 2^k елементів. Суперечність завершує доведення достатності.

Двоїстою до теореми 3.5.3 є така теорема:

ТЕОРЕМА 3.5.4. *Нехай A — формула логіки предикатів, яка містить k одномісних предикатних змінних і не має ніяких інших предикатних змінних. Для того щоб формула A була виконуваною, необхідно і достатньо, щоб формула A була виконуваною на множині, яка містить рівно 2^k елементів.*

Достатність — очевидна.

Необхідність. Нехай A — виконувана формула. Тоді $\urcorner A$ не є логічно загальнозначущою, отже за Т3.5.3 формула $\urcorner A$ не є загальнозначущою на множині, що містить точно 2^k елементів. Тому її заперечення, тобто $\urcorner\urcorner A \equiv A$ — формула виконувана на множині з 2^k елементів.

ТЕОРЕМА 3.5.5. *Нехай A $(x_1, ..., x_m)$ — довільна відкрита формула логіки предикатів. Формула*

$$\forall x_1, ..., \forall x_m\, A\, (x_1, ..., x_m) \tag{3.5.2}$$

є логічно загальнозначуща тоді і тільки тоді, коли вона загальнозначуща на множині з т елементів.

Необхідність — очевидна.

Достатність. Формула (3.5.2) — загальнозначуща на множині з m елементів. Припустимо, що (3.5.2) не є логічно загальнозначуща. Тоді знайдеться множина M_0 з числом елементів, більшим m, відповідні предикати, визначені на M_0 і такі елементи $a_1, ..., a_m$ з множини M_0, що інтерпретація $A^*(a_1, ..., a_m)$ є хибне висловлення. Це суперечить загальнозначущості (3.5.2) в області, яка містить m елементів.

ТЕОРЕМА 3.5.6*. Нехай $A(x_1, ..., x_n)$ — довільна відкрита формула логіки предикатів. Формула*

$$\exists x_1, ..., \exists x_n A(x_1, ..., x_n) \qquad (3.5.3)$$

є логічно загальнозначущою тоді і тільки тоді, коли вона загальнозначуща на одноелементній множині.

Необхідність — очевидна.

Достатність. Припустимо, що (3.5.3) не є логічно загальнозначуща. Тоді й поготів формула $\exists x A(x, ..., x)$ не буде логічно загальнозначущою. Ця формула сильніше твердження ніж (3.5.3). Отже, існує множина M_1, таке $a \in M_1$, і конкретні предикати, визначені на M_1, такі, що одержана інтерпретація $A(a, ..., a)$ є хибним висловленням, що суперечить умові загальнозначущості A на одноелементній множині.

Вправи

1. Написати безкванторну формулу, логічно еквівалентну на двоелементній множині $\{a, b\}$ формулі логіки предикатів:

a) $\forall x \forall y \forall z \mathrm{P}(x, y, z)$;
b) $\exists x \forall y \mathrm{P}(x, y) \vee \mathrm{Q}(x)$;
c) $\exists x \forall y \mathrm{P}(x, x) \wedge \rceil \mathrm{P}(x, y)$;
d) $\forall x \exists y \mathrm{P}(x, y) \vee \exists y \forall x \rceil \mathrm{P}(x, y)$.

2. Перевірити, чи є формула лзз:

$$\forall x \exists y \mathrm{P}(x, y) \vee \exists y \forall x \mathrm{P}(x, y).$$

3. Довести твердження.

Нехай $A(x, y)$ – формула логіки предикатів, яка не містить символів кванторів. Формула $\forall x \forall y A(x, y)$ є лзз тоді і тільки тоді, коли вона є загальнозначущою на одноелементній множині.

4. Довести твердження.

Нехай $A(x)$ — формула логіки предикатів, яка не містить символів кванторів. Формула $A(x)$ є лзз тоді і тільки тоді, коли вона є загальнозначущою на двоелементній множині.

3.6. ЗАСТОСУВАННЯ СИМВОЛІКИ МАТЕМАТИЧНОЇ ЛОГІКИ В МАТЕМАТИЧНИХ ФОРМУЛЮВАННЯХ

Практично найважливішою сферою застосування логіки предикатів до логіко-математичної практики є сфера побудови доведень різних теорем. Основну увагу приділимо застосуванню тих аспектів логіки предикатів, які корисні вчителю математики не тільки для формування його логічної культури, але і для практичного оперування розглядуваними поняттями і методами безпосередньо у процесі викладання: грамотний запис на мові логіки предикатів різних математичних речень; більш поглиблений, ніж в алгебрі висловлень, аналіз будови математичних теорем; застосування логіки предикатів у теорії множин.

В математиці часто зустрічаються вирази типу "принаймні n", "не більш ніж n", "n і тільки n", тощо, де n — натуральне число.

Ці вирази називають числовими кванторами. Вони мають чисто логічний зміст, тому що їх можна виразити без числівників на мові кванторів загальності та існування, логічних операцій над предикатами та знака $\equiv$, що позначає тотожність (співпадання) об'єктів.

Речення "Принаймні один об'єкт має властивість P" має той же зміст, що і речення "Існує об'єкт, що має властивість P", тобто

$$\exists x P(x). \tag{3.6.1}$$

Речення "не більш ніж один об'єкт має властивість P" рівнозначне за змістом реченню "Якщо є об'єкти, що мають властивість P, то вони співпадають", тобто

$$\forall x \forall y [P(x) \wedge P(y) \Rightarrow x = y]. \tag{3.6.2}$$

Нарешті, речення "Один і тільки один об'єкт має властивість P" рівнозначне кон'юнкції висловлень (3.6.1) і (3.6.2):

$$\exists x P(x) \wedge \forall x \forall y [P(x) \wedge P(y) \Rightarrow x = y]. \tag{3.6.3}$$

Співставлення одновимірному предикату $P(x)$ висловлення (3.6.3) іноді позначають так:$\exists! x P(x)$.

Символ $\exists! x$ називають квантором існування та єдиності по змінній x.

Приклад 3.6.1. Використовуючи цей квантор, запишемо висловлення: "Будь-яка збіжна послідовність має рівно одну границю":

$$\forall a_n\ \exists a(a = \lim a_n) \Rightarrow \exists! a(a = \lim a_n).$$

Змінна під знаком квантора часто пробігає не всю універсальну множину, а деяку його підмножину. У математиці універсальною числовою множиною є множина всіх дійсних чисел R, але часто вживаються вирази "для кожного $\varepsilon > 0$", "існує $\delta > 0$", де ε і δ набувають значень на множині додатних чисел. Тому вводять символи $\underset{M}{\exists x}, \underset{M}{\exists y}$, де під символом кванторної змінної пишуть назву тієї підмножини *M*, яку пробігає ця змінна .

Символи $\underset{M}{\exists x}, \underset{M}{\exists y}$ називаються *обмеженими кванторами.*

ОЗНАЧЕННЯ 3.6.1*. Обмеженим квантором* $\underset{M}{\forall x}\, P(x)$ *назвемо вираз, який є скороченням виразу* $\forall x(x \in M \Rightarrow P(x))$, *тобто* $\underset{M}{\forall x}\, P(x) =^{df} \forall x(x \in M \Rightarrow P(x))$.

Обмеженим квантором $\underset{M}{\exists x}\, P(x)$ *назвемо вираз, який є скороченням виразу* $\exists x(x \in M \wedge P(x))$, *тобто* $\underset{M}{\exists x}\, P(x) = \exists x(x \in M \wedge P(x))$.

Якщо в деякій рівносильності, яка не містить символів вільних предметних змінних, замінити всі квантори $\forall x$, $\exists x$ символами обмежених кванторів $\underset{M}{\forall x}, \underset{M}{\exists x}$, відповідно, де $M \neq \varnothing$, то одержимо знову рівносильність, тобто тотожно істинну формулу.

Символіка математичної логіки допомагає кращому засвоєнню логічної структури математичних понять, більший стислості і чіткості відповідних формулювань і зручності оперування з ними. Теорія множин є найближчою до логіки математичною дисципліною. Рівність множин *A* і *B* визначається співвідношенням:

1^0. $A = B \Leftrightarrow \forall x(x \in A \Leftrightarrow x \in B)$.

Поняття підмножини:

2^0. $A \subset B \Leftrightarrow \forall x(x \in A \Rightarrow x \in B)$.

Операції об'єднання, перетину, доповнення і віднімання:

3^0. $x \in A \cup B \Leftrightarrow x \in A \vee x \in B$;
4^0. $x \in A \cap B \Leftrightarrow x \in A \wedge x \in B$;
5^0. $x \in CA \Leftrightarrow \rceil(x \in A)$;
6^0. $x \in A \setminus B \Leftrightarrow x \in A \wedge \rceil(x \in B)$.

З'ясуванню логічної структури математичних понять та ролі логічних операцій в їх формулюванні допомагає запис за допомогою логіко-математичної символіки.

За допомогою кванторної символіки зручно записувати формулювання різних означень та теорем, у процесі такого запису приходиться осмислювати дане речення, чітко виділяти в ньому посилки і висновок (якщо це теорема), окреслювати більш широке коло понять і чітко виділяти обмежуючу умову (якщо це означення). Одним словом, переклад розпливчатого словесного формулювання на строгу мову логіки предикатів, що не опускає суперечливих тлумачень, сприяє чіткості і ясності мислення. Розглянемо деякі приклади.

Запишемо в символічній мові означення функції $f(x)$, обмеженої на даній множині.

$f(x)$ — обмежена на $M =^{\mathrm{df}} \underset{R+}{\exists a} \underset{M}{\forall x} (\,|f(x)| \le (a)$.

У визначальній частині чітко виділяється роль кожного квантора і порядок їх слідування. Якщо, наприклад, в приставці правої частини замінити порядок кванторів на $\underset{M}{\forall x} \underset{R+}{\exists a}$, то дістанемо означення зовсім іншого поняття, а саме функції, скінченої на множині M.

Відміна в логічній структурі понять нескінченно великої і необмеженої послідовності стає цілком прозорою, коли перейти до логічної символіки.

Послідовність (x_n) — нескінченно велика $=^{\mathrm{df}} \underset{R}{\forall p} \underset{N}{\exists m} \underset{N}{\forall n}$, $(n > m) \Rightarrow |x_n| > p)$.

Послідовність (x_n) — необмежена $=^{\mathrm{df}} \underset{R}{\forall p} \underset{N}{\forall m} \underset{N}{\exists n}$ $(n > m)$ $\wedge |x_n| > p)$.

Відміна між поняттям неперервності функції f на множині M і рівномірної неперервності на множині M при символічному записі зводиться до перестановки кванторів – квантора загальності і квантора існування.

Функція f — неперервна на M:

$$M =^{\mathrm{df}} \underset{R+}{\forall \varepsilon} \underset{M}{\forall x} \underset{M}{\forall y} \underset{R+}{\exists \delta} (\,|x - y| < \delta \Rightarrow |f(x) - f(y)| < \varepsilon).$$

Функція f — рівномірно неперервна на М:

$$M =^{\mathrm{df}} \underset{R+}{\forall \varepsilon} \underset{R+}{\exists \delta} \underset{M}{\forall x} \underset{M}{\forall y} (\,|x - y| < \delta \Rightarrow |f(x) - f(y)| < \varepsilon).$$

Те, що з рівномірної неперервності функції на M випливає неперервність її на M є наслідком теореми логіки: $\exists y \forall x P(x, y) \Rightarrow \forall x \exists y P(x, y)$.

Тобто перестановка двох різнойменних кванторів у формулі може привести до утворення нового поняття. Перестановка однойменних кванторів, які безпосередньо слідують один за одним, приводить до логічно еквівалентного твердження. Застереження про те, що однойменні квантори, які переставляються, повинні безпосередньо слідувати один за одним, є дуже істотне.

Порівняємо логічну структуру означень найбільшого елемента частково упорядкованої множини та її максимального елемента.

1) y — найбільший елемент $D =^{\mathrm{df}} \forall x(x \in D \Rightarrow x \le y)$;

2) y — максимальний елемент $D =^{\mathrm{df}} \rceil \exists x(x \in D \wedge x > y)$.

Перетворюючи дефініенс 2) за правилами логіки, дістанемо рівнозначний вираз

$$\forall x\rceil(x\in D\wedge x>y)\Leftrightarrow\forall x(\rceil(x\in \mathrm{D})\vee\rceil(x>y)) \Leftrightarrow \forall x(x\in D\Rightarrow\rceil(x>y)).\ (*)$$

Твердження (*) нерівнозначне дефініенсу 1), останнє сильніше. Дійсно, у випадку часткового впорядкування множини з цього слідує висловлення (*) але не навпаки: з $\rceil(x > y)$ може слідувати не тільки $x \le y$, а також, що x не дорівнює y.

У процесі математичних міркувань часто приходиться оперувати запереченням. Візьмемо доведення від супротивного або доведення заперечних тверджень.

Необхідно довести що $a \ne sup\ M$ (тобто a не є точною верхньою межею).

$$(a = sup\ M) =^{\mathrm{df}} \forall x(x \le a) \wedge \forall\varepsilon\exists x(x > a - \varepsilon). \qquad (3.6.4)$$

1. $a \ne sup\ M$;
2. $\rceil(\forall x\ (x \le a) \wedge \forall\varepsilon\exists x(x > a - \varepsilon))$ (1, (6.4));
3. $\rceil(\forall x(x \le a)) \vee \rceil(\forall\varepsilon\exists x(x > a\text{-}\varepsilon))$ (2, закон де Моргана);
4. $\exists x(x > a) \vee \exists\varepsilon\forall x(x \le a - \varepsilon)$ (3, закон де Моргана для кванторів).

Тобто, для доведення того, що дійсне число a не є $sup\ M$, необхідно довести, що a не є верхньою межею, або, що існує число, менше від a, яке є верхньою межею M.

У процесі математичних доведень часто зустрічаються підстановки математичних висловлень замість змінних. Ці підстановки повинні задовольняти обмеженням логічного характеру.

1. Підставляти можна тільки замість вільних змінних.
2. Жодне вільне входження змінної внаслідок підстановки не повинно стати зв'язаним.

Ці обмеження стосуються не тільки змінних зв'язаних кванторами, але й змінних зв'язаних математичними операторами.

Прийнято для скорочення опускати квантори загальності, якщо вони знаходяться на початку формули. Квантор існування опускати не можна. Це може призвести до грубих помилок.

Вправи

1. Використовуючи теорему про знак квадратного тричлена, визначити, чи є істинним твердження:

a) $\forall p\forall q(\forall x(x^2 + px + q > 0) \Rightarrow q > 0)$;
b) $\forall p\forall q\forall x(x^2 + px + q > 0 \Rightarrow q > 0)$.

2. Записати логіко-математичною символікою теорему Вієта для всіх квадратних рівнянь виду $x^2 + px + q = 0$. Чи буде правильним такий запис теореми Вієта:

$\forall p\forall q\forall a\forall b(\forall x(x^2+px+q>0) \Rightarrow x = a\vee x = b \Rightarrow p = -(a+b)\wedge q = ab)$;

3. Визначити, чи є істинними на множині всіх дійсних чисел такі математичні твердження:

a) $\forall x \forall y \exists z (z = \sqrt{x^2 + y^2})$;

b) $\forall x \exists z \forall y (z = \sqrt{x^2 + y^2})$;

c) $\forall x \forall y (x^2 + y^2 = 0 \Rightarrow x = 0 \wedge y = 0$;

d) $\forall x \exists y (y = 10^x \wedge \underset{D+}{\forall y} \exists x (y = 10^x)$.

e) $\forall x \forall y \exists z (z = \ln(x^2 + y^2))$;

f) $\forall x \forall y (\ln(xy) = \ln x + \ln y)$;

g) $\forall x \exists y (xy = 1)$.

4. Оцінити істинність математичних тверджень на множині всіх дійсних чисел:

c) $\forall x \forall y \underset{C}{\forall k} (tgx = tgy \Rightarrow y = x + k\pi)$;

d) $\forall x \forall y \underset{C}{\exists k} (tgx = tgy \Rightarrow y = x + k\pi)$;

e) $\forall x \forall y \underset{C}{\exists k} (tgx = tgy \Leftrightarrow y = x + k\pi)$;

f) $\forall x \forall y ((tgx = tgy) \Rightarrow \underset{C}{\exists k} (y = x + k\pi))$.

5. Записати логіко-математичною символікою хід розв'язання рівнянь:

a) $\sqrt{x-1} = x - 3$;

b) $\sqrt{x+3} = x + 1$;

c) $\lg^2 10x + \lg x = 19$.

6. Записати логіко-математичною символікою хід розв'язання систем рівнянь:

a) $\begin{cases} x + 2y = 7, \\ 3x + y = 6; \end{cases}$

b) $\begin{cases} x + y = 1, \\ y^2 = y; \end{cases}$

c) $\begin{cases} x - y = 3, \\ x^2 - 2y^2 = 14. \end{cases}$

Розділ 4

ЧИСЛЕННЯ ПРЕДИКАТІВ

4.1. ПОБУДОВА ЧИСЛЕННЯ ПРЕДИКАТІВ (ТЕОРІЇ PL)

У звичайних аксіоматичних математичних теоріях усі міркування проводяться в звичайній мові, логічні засоби, які застосовуються при цьому, не описуються точно, а вважаються інтуїтивно зрозумілими.

У формалізованих теоріях в аксіомах описуються точно властивості первинних понять, і аналогічно визначається мова теорії, завдяки чому можна точно визначити сам процес логічної дедукції.

Розглянемо побудову числення предикатів — теорії PL.

Алфавіт теорії — початкові символи складається з:

1) зчисленної множини символів $x_1, ..., x_n, ...$ які називаються індивідними змінними;

2) скінченної (можливо порожньої) або зчисленної множини $a_1, ..., a_n, ...$ які називають індивідними сталими (константами);

3) скінченної (можливо порожньої) або зчисленної множини символів f_i^j $(i = 1, 2, ..., j = 1, 2, ...)$, які називаються функціональними літерами;

4) скінченної непорожньої або зчисленної множини символів P_i^j $(i = 1, 2, 3, ..., j = 0, 1, 2, 3, ...)$, які називають предикатними літерами.

5) трьохелементної множини логічних символів $\{\urcorner, \Rightarrow, \forall\}$;

6) чотирьохелементної множини розділових знаків {(,), ", "}.

Терм є аналогом назви предметів змістовної теорії, формули теорії PL інтерпретуються як її твердження.

ОЗНАЧЕННЯ 4.1.1.

1) кожне x_k і кожна a_k є термами теорії PL $(k \in N)$;

2) якщо $t_1, ..., t_n$ — терми, то $f_i^n(t_1, ..., t_n)$ — терм $(i \in N)$;

3) ніяких інших термів, крім утворених за п.п. 1), 2), немає.

ОЗНАЧЕННЯ 4.1.2.

1) якщо $t_1, ..., t_n$ — терми теорії, то вираз $P_i^n(t_1, ..., t_n)$ — формула, яку називають елементарною;

2) якщо A і B — формули теорії, то вирази $(\neg A)$, $(A \Rightarrow B)$, $(\forall x_i A)$ є теж її формули;

3) жодних інших формул теорії, які б не можна було дістати за п.п. 1), 2), немає.

Введений алфавіт визначає мову, яка називається мовою першого порядку, а відповідну формальну теорію — теорією першого порядку.

Для скорочення числа дужок, залишимо в силі умови, прийняті щодо цього в численні висловлень і введемо через відповідні означення в метамові символи $\exists$, $\wedge$, $\vee$, $\Leftrightarrow$.

$$A \vee B = \neg A \Rightarrow B,\ A \wedge B = \neg(A \Rightarrow \neg B);\ A \Leftrightarrow B = (A \Rightarrow B) \wedge (B \Rightarrow A),$$
$$\exists x(A) = \neg \forall x(\neg A).$$

Аксіоми теорії *PL*

L1. $\boldsymbol{A \Rightarrow (B \Rightarrow A)}$.

L2. $\boldsymbol{(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))}$.

L3. $\boldsymbol{(\neg A \Rightarrow \neg B) \Rightarrow ((\neg A \Rightarrow B) \Rightarrow A)}$.

L4. $\boldsymbol{\forall x_i\, A(x_i) \Rightarrow A(t)}$, якщо терм t вільний для x_i в A.

L5. $\boldsymbol{\forall x_i (A \Rightarrow B) \Rightarrow (A \Rightarrow \forall x_i B)}$ якщо x_i не має вільних входжень в A.

Для теорії *PL* мають місце два ***правила виведення***:

1. З A та $A \Rightarrow B$ виводиться B (*правило виведення — MP*).
2. З A виводиться $\forall x_i A$ (*правило введення квантора $\forall$ — B$\forall$*).

Через A і B позначені довільні формули теорії *PL*.

ОЗНАЧЕННЯ 4.1.3. *Скінчена послідовність формул теорії PL*

$$B_1,\ \ldots,\ B_n \tag{4.1.1}$$

називається доведенням (формальним) формули B, якщо B_n співпадає з B і кожний член послідовності (4.1.1) є або аксіомою, або безпосередньо виводиться з попередніх членів послідовності (4.1.1) за правилом MP або B $\forall$.

Формула теорії *PL*, для якої існує доведення, називається *теоремою теорії T* і позначається $\vdash B$.

ОЗНАЧЕННЯ 4.1.4. *Скінчена послідовність формул теорії PL*

$$B_1,\ \ldots,\ B_n \tag{4.1.2}$$

називається виведенням B з гіпотез $A_1, \ldots, A_m$, якщо B_n співпадає з B і кожний член послідовності (4.1.2) є або аксіомою, або однією з гіпотез $A_1, \ldots, A_m$, або виводиться з попередніх членів послідовності (4.1.2) за правилом MP або B$\forall$. В цьому випадку кажуть, що B виводиться з $A_1, \ldots, A_m$, що позначається $A_1, \ldots, A_m \vdash B$.

Вправи

1. Побудувати виведення формули A з формули $\forall x \forall y A$, де A — довільна формула.

2. Довести:

a) $\vdash \forall x P(x) \Rightarrow \forall x P(x)$;
b) $\vdash \exists x Q(x) \Leftrightarrow \exists x Q(x)$;
c) $\vdash \exists x P(x) \Rightarrow (Q(x) \Rightarrow \exists x P(x)$;
d) $\vdash P(x) \Rightarrow P(x) \vee \exists x Q(x) \wedge \neg P(x)$;
e) $\vdash (P(x) \wedge \neg Q(x)) \wedge (\neg R(x) \Rightarrow P(x)) \Rightarrow Q(x) \wedge \neg P(x)$.

3. Побудувати виведення:

$$\forall x (P(x) \Rightarrow Q(x)), \forall x P(x) \vdash \forall x Q(x).$$

4.2. МЕТАТЕОРЕМА ДЕДУКЦІЇ

Доведемо твердження, яке дозволить використовувати в теорії PL тавтології алгебри висловлень.

ОЗНАЧЕННЯ 4.2.1. *Окремим випадком тавтології A в теорії PL назвемо формулу теорії, яка утворюється з A підстановкою замість кожної пропозиційної букви, що входить в A певної формули теорії PL.*

МЕТАТЕОРЕМА 1. *Якщо формула B теорії PL є окремим випадком тавтології, то $\vdash B$, причому останнє доведення можна отримати використовуючи тільки аксіоми L1–L3 і правило MP.*

Доведення. Нехай B — окремий випадок тавтології A. Тоді за властивістю повноти числення висловлень формула A буде теоремою числення висловлень, тобто існує доведення A, а саме певна послідовність формул числення висловлень

$$B_1, \ldots, B_n \tag{4.2.1}$$

де B_n співпадає з A. Причому при доведенні застосовуються тільки аксіоми L1–L3 і правило MP.

Підставимо замість кожної пропозиційної літери в (4.2.1), ту формулу теорії PL, якою ця літера заміщалася при утворенні B з A, а замість кожної пропозиційної літери, що не входить в B, підставимо довільну, але одну і ту ж формулу теорії PL. Утворена послідовність формул і є доведенням B в теорії PL, причому при доведенні використано тільки L1–L3 і правило MP.

Теорема дедукції справджується для теорії PL, якщо в її формулювання ввести певне застереження. Розглянемо деяке виведення формули B теорії PL з гіпотез $A_1, \ldots, A_n$, а саме $B_1, \ldots, B_m$ разом з аналізом цього виведення.

ОЗНАЧЕННЯ 4.2.2. *Формула B_i називається залежною від гіпотези A_k тоді і тільки тоді, коли:*

1) B_i співпадає з A_k, причому цим обґрунтовується в аналізі входження B_i в виведення, або

2) B_i є безпосереднім висновком формул, з яких хоча б одна залежить від A_k.

ЛЕМА. *Якщо $\Gamma, A \vdash B$ і існує таке виведення $B_1, ..., B_m = B$, в якому жодна з формул B_i не залежить від A, то $\Gamma \vdash B$.*

Доведення. Нехай

$$B_1, ..., B_m \tag{4.2.2}$$

дане виведення формули B з $\Gamma \cup \{A\}$. Доведемо за індукцією по довжині m виведення (4.2.2). За індуктивним припущенням, яке ми приймемо, лема справджується при кожному $j < i$. Доведемо, що вона справедлива для i.

За означенням вивідності з гіпотез, для довільного $i \leq m$ B_i є: 1) або аксіомою теорії PL, 2) або $B_i \in \Gamma$, 3) або $B_i = A$, 4) або B_i безпосередньо виводиться з попередніх членів (4.2.2) за правилами MP чи В$\forall$.

У 1) або 2) випадках, відразу $\Gamma \vdash B_i$ (за означенням вивідності).

Випадок 3) відпадає за умовою леми (жодна з B_i не залежить від A).

Якщо випадок 4), то за індуктивним припущенням для кожного $j < i$ виконується $\Gamma \vdash B_j$, а тоді виконуватиметься $\Gamma \vdash B_i$ (за означенням вивідності).

Згідно принципу індукції доводжуване твердження справджується при всіх натуральних i, зокрема при $i = m$.

Нехай A, B — довільні формули, а Γ — множина формул теорії PL.

МЕТАТЕОРЕМА ДЕДУКЦІЇ. *Нехай $\Gamma, A \vdash B$, причому існує таке виведення формули B з $\Gamma; A$, в якому правило В$\forall$ не застосовується до жодної формули, залежної від A по тій змінній x_i, яка має вільне входження в A. Тоді*

$$\Gamma \vdash A \Rightarrow B.$$

Доведення. Дано, що існує виведення (4.2.2) формули B з $\Gamma \cup \{A\}$. У послідовності (4.2.2) для кожного $i \leq m$ B_i є: 1) або аксіомою теорії PL; 2) або $B_i = A$. 3) або $B_i \in \Gamma$; 4) або існують $j < i$; $k < i$ такі, що $B_j = B_k \Rightarrow B_i$ 5) або існує таке $j < i$, що $B_i = \forall x_k B_j$.

Доведення проведемо за індукцією по i. Індуктивне припущення — теорема справджується для всіх $s < i$. Покажемо, що тоді у всіх випадках 1)-5) вона справджується при $s = i$. У випадках 1)-4) доведення проводиться як і у численні висловлень.

5) Тут правило В$\forall$ застосовується до формули B_j по змінній x_k, тому згідно з умовою:

I) B_j не залежить від A, або II) $x_к$ не має вільних входжень в A.

5_I) Нехай $B_i = \forall x_к\, B_j$, де $j < i$ та B_j не залежить від A. За індуктивним припущенням $Г \vdash A \Rightarrow B_j$. Отже $Г, A \vdash B_j$, звідки за лемою $Г \vdash B_j$, оскільки B_j не залежить від A. Тоді за правилом В$\forall$ матимемо $Г \vdash \forall x_к B_j$, тобто $Г \vdash B_i$. Далі з B_i виводиться $A \Rightarrow B_i$ (за L1 і *MP*). Отже, з індуктивного припущення випливає $Г \vdash A \Rightarrow B_i$.

5_{II}) Нехай $B_i = \forall x_к\, B_j$ та $x_к$ не має вільного входження в A. За індуктивним припущенням $Г \vdash A \Rightarrow B_j$. Звідси за правилом В$\forall$ одержимо $Г \vdash \forall x_к(A \Rightarrow B_j)$. За схемою аксіом L5 маємо $\forall x_к(A \Rightarrow B_j) \Rightarrow (A \Rightarrow \forall x_к B_j)$ оскільки $x_к$ не входить як вільна змінна в A, і формула виводиться з $Г$. Враховуючи, що $Г \vdash \forall x_к(A \Rightarrow B_j)$, за правилом *MP* маємо $Г \vdash A \Rightarrow \forall x_к B_j$, тобто як і у попередньому випадку.

Отже у випадках 1)-5) з індуктивного припущення випливає, що доводжуване твердження справджується й при $j = i$. За принципом індукції воно справедливе при всіх $i \leq m$.

НАСЛІДОК 1. *Якщо $Г, A \vdash B$ та A — замкнена формула, то $Г \vdash A \Rightarrow B$.*

НАСЛІДОК 2. *Якщо $Г, A \vdash B$ та існує виведення B з $Г, A$, в якому не застосовується правило В$\forall$, то $Г \vdash A \Rightarrow B$.*

Доведемо теорему числення предикатів.

Теорема 1. $\forall x_1(A \Rightarrow B) \Rightarrow (\forall x_1 A \Rightarrow \forall x_1 B)$.

1. $\forall x_1(A \Rightarrow B)$	гіпотеза;
2. $\forall x_1 A$	гіпотеза;
3. $\forall x_1(A \Rightarrow B) \Rightarrow (A \Rightarrow B)$	L4;
4. $A \Rightarrow B$	MP(1, 3);
5. $\forall x_1 A \Rightarrow A$	L4;
6. A	MP(2, 5);
7. B	MP(4, 6);
8 $\forall x_1 B$	В$\forall$(7).

Тобто $\forall x_1(A \Rightarrow B), \forall x_1 A \vdash \forall x_1 B$ (1-8, за означенням вивідності).

Оскільки правило В$\forall$ застосоване по змінній x_1 яка не входить вільно ні у $\forall x_1(A \Rightarrow B)$, ні у $\forall x_1 A$, то, застосовуючи МТД два рази, одержимо $\forall x_1(A \Rightarrow B) \Rightarrow (\forall x_1 A \Rightarrow \forall x_1 B)$.

Застосовуючи метатеорему 1 про окремі випадки тавтологій, можна обґрунтувати перенесення на теорію *PL* числення предикатів, ряду похідних правил виведення з числення висловлень.

Правило ВвД: $A \vdash A \vee B$; $B \vdash A \vee B$.
ВвК: $A, B \vdash A \wedge B$.
ВД: $A \Rightarrow C, B \Rightarrow C \vdash A \vee B \Rightarrow C$.
ВК: $A \wedge B \vdash A$; $A \wedge B \vdash B$.
КП: $A \Rightarrow B \vdash \neg B \Rightarrow \neg A$.
ПС: $A \Rightarrow B, B \Rightarrow C \vdash A \Rightarrow C$.

Доведення правила ВвК.

Нехай R — довільна аксіома теорії. Запишемо послідовність формул:

1. A	гіпотеза;
2. B	гіпотеза;
3. $(R \Rightarrow A) \Rightarrow ((R \Rightarrow B) \Rightarrow (R \Rightarrow A \wedge B))$	тавтологія;
4. $A \Rightarrow (R \Rightarrow A)$	L1;
5. $R \Rightarrow A$	MP(1, 4);
6. $(R \Rightarrow B) \Rightarrow (R \Rightarrow A \wedge B)$	MP(3, 5);
7. $B \Rightarrow (R \Rightarrow B)$	L1;
8. $R \Rightarrow B$	MP(2, 7);
9. $R \Rightarrow A \wedge B$	MP(8, 6);
10. R	аксіома;
11. $A \wedge B$	MP(1, 9).

Отже, $A, B \vdash A \wedge B$ — за означенням вивідності з послідовності формул 1-11.

Доведення решти похідних правил пропонується провести самостійно.

Вправи

1. Нехай A не містить x вільно. Довести вивідність:

a) $\vdash (\forall x A \Leftrightarrow A)$;
b) $\vdash (\exists x A \Leftrightarrow A)$;
c) $\vdash (\forall x \forall y B(x, y) \Leftrightarrow \forall y \forall x B(x, y))$;
d) $\vdash (\exists x \exists y B(x, y) \Leftrightarrow \exists y \exists x B(x, y))$;
e) $\vdash (\forall x \forall y B(x, y) \Rightarrow \forall x B(x, x))$;
f) $\vdash (\exists x B(x, x) \Rightarrow \exists y \exists x B(x, y))$;
g) $\vdash (\exists x B(x) \Rightarrow \neg \forall x \neg B(x))$;
h) $\vdash (\forall x B(x) \Rightarrow \neg \exists x \neg B(x))$;
i) $\vdash (\neg \exists x B(x) \Rightarrow \forall x \neg B(x))$.

2. Побудувати виведення формул:

a) $\vdash (\forall x \forall y B(x, y) \Rightarrow \forall y \forall x B(x, y))$;
b) $\vdash (\exists x \exists y B(x, y) \Leftrightarrow \exists y \exists x B(x, y))$;
c) $\vdash (\exists x \forall y B(x, y) \Leftrightarrow \forall y \exists x B(x, y))$.

4.3. ІНТЕРПРЕТАЦІЯ ТЕОРІЇ PL

Розглянемо математичну структуру M, тобто не порожню множину M^*, на якій визначені n-арні відношення $R_i(x_1, ..., x_n)$ і m-арні операції.

ОЗНАЧЕННЯ 4.3.1. *Інтерпретацією формули A теорії PL у структурі M назвемо систему, що складається з множини M* яку називають областю інтерпретації і відображення, за яким кожній n-арній предикатній літері в A відповідає n-арне відношення в M, кожній m-арній функціональній літері в A відповідає m-арна операція в M і кожній предметній сталій в A — назва певного елемента множини M*.*

Будемо вважати, що елементарна формула теорії $P_i^n(t_1, ..., t_n)$ в даній інтерпретації на структурі M при даній заміні всіх її вільних індивідних змінних назвами елементів M^* має значення "істина" (1) тоді і тільки тоді, коли відповідне n-арне відношення $R(t_1', ..., t_n')$ виконується в структурі M.

При даній інтерпретації на структурі M і певному заміщенні всіх вільних індивідних змінних назвами елементів M^*:

- формула $\urcorner A$ набуває значення "істина" (1) тоді і тільки тоді, коли A набуває значення (0);
- формула $A \wedge B$ набуває значення "істина" (1) тоді і тільки тоді, коли обидві формули A і B набувають значення (1);
- формула $A \vee B$ набуває значення "істина" (1) тоді і тільки тоді, коли хоча б одна з формул набуває значення (1);
- формула $A \Rightarrow B$ набуває значення "істина" (1) тоді і тільки тоді, коли B набуває значення (1) або A приймає значення (0);
- формула $\forall x_i A(x_i)$ набуває значення "істина" тоді і тільки тоді, коли для кожного елемента $b \in M^*$ $A(b)$ має значення (1);
- формула $\exists x_i A(x_i)$ набуває значення "істина" тоді і тільки тоді, коли існує хоча б один елемент b множини M^* такий, що $A(b)$ набуває значення (1).

При інтерпретації замкненої формули теорії PL одержимо висловлення (істинне або хибне), тоді як формула, в якій є вільні входження індивідних змінних, інтерпретується як висловлювальна форма.

ОЗНАЧЕННЯ 4.3.2. *Формула A називається істинною в даній інтерпретації, якщо вона приймає значення "істина" в цій інтерпретації при всіх замінах вільних змінних в A назвами елементів області інтерпретації.*

ОЗНАЧЕННЯ 4.3.3. *Формула A називається логічно загальнозначущою, якщо вона істинна в кожній інтерпретації.*

ОЗНАЧЕННЯ 4.3.4. *Інтерпретація в якій всі формули даної множини формул Γ є істинними, називається моделлю даної множини Γ.*

Нехай дано дві інтерпретації I_1 і I_2 множини формул $Γ$ на структурах M_1 і M_2, причому в M_1 і M_2 інтерпретаціями предикатних літер P_i є відношення R_i і R_i', інтерпретаціями функціональних літер f_i є символи операцій F_i і F_i', а індивидні сталі a_i інтерпретуються як c_i і c_i' відповідно.

ОЗНАЧЕННЯ 4.3.5. *Інтерпретації I_1 і I_2 називаються ізоморфними, якщо існує взаємно-однозначне відображення φ множини M_1^* на M_2^*, таке, що виконуються твердження:*

1) для довільних $b_1, ..., b_n \in M_1^*$ $R_i(b_1, ..., b_n)$ виконується в M_1 тоді і тільки тоді, коли $R_i'(\varphi(b_1), ..., \varphi(b_n))$ виконується в M_2;

2) $\varphi(F_i(b_1, ..., b_m)) = F_i'(\varphi(b_1), ..., \varphi(b_m))$;

3) $\varphi(c_i) = c_i'$ $(i = 1, 2, ...)$.

Оскільки φ — взаємно-однозначне відображення M_1^* на M_2^* то, очевидно необхідною (але не достатньою) умовою ізоморфності інтерпретацій є рівнопотужність областей інтерпретацій I_1 і I_2.

4.4. ПИТАННЯ НЕСУПЕРЕЧНОСТІ, ПОВНОТИ, ПРОБЛЕМА ВИРІШЕННЯ

ОЗНАЧЕННЯ 4.4.1. *Теорія PL називається внутрішньо або просто несуперечною якщо не існує такої формули A, вираженої в термінах теорії PL, що A і $\neg A$ є теоремами теорії PL.*

Для довільної формули A позначимо через $W(A)$ вираз, який одержується внаслідок опускання в A всіх символів термів, кванторів, відповідних дужок і ком.

З означення W безпосередньо випливає:

$$W(\neg A) = \neg W(A); \tag{4.4.1}$$

$$W(A \Rightarrow B) = W(A) \Rightarrow W(B). \tag{4.4.2}$$

МЕТАТЕОРЕМА 2*. Якщо $\vdash B$, то $\vdash W(B)$.*

Оскільки B — теорема числення предикатів, то існує скінчена послідовність формул цього числення

$$B_1, ..., B_n, \tag{4.4.3}$$

така що $B_n = B$, B_i є: 1) або схемою аксіом $L1–L5$; 2) або виводиться з попередніх членів послідовності (4.4.3) у випадку

2_1) за *MP*; у випадку 2_2) за В∀. Індуктивне припущення — доводжуване твердження справджується для всіх B_s де $s<i$. Покажемо, що тоді воно справджується також і для B_i.

1) Якщо B_i — окремий випадок L1−L3, то B_i — тавтологія, а $W(B_i)$ співпадає з B_i.

Якщо B_i — окремий випадок L4, то

$$W(B_i) = A \Rightarrow A. \tag{4.4.4}$$

Якщо B_i — окремий випадок L5, то

$$W(B_i) = (A \Rightarrow B) \Rightarrow (A \Rightarrow B), \tag{4.4.5}$$

де A та B — формули числення висловлень. Формули (4.4.4) та (4.4.5) — тавтології.

2_1) існують такі $j < i$, $k < i$, що $B_к = B_j \Rightarrow B_i$. За індуктивним припущенням $W(B_к)$ та $W(B_j)$ — тавтології. Але $W(B_к) = W(B_j \Rightarrow B_i) = W(B_j) \Rightarrow W(B_i)$ (згідно із співвідношенням (4.4.2)). Оскільки $W(B_j)$ та $W(B_j) \Rightarrow W(B_i)$ тавтології, то $W(B_i)$ — теж тавтологія.

2_2) існує таке $k < i$, що $B_i = \forall x_p B_к$. За індуктивним припущенням $W(B_к)$ – тавтологія, а $W(B_i) = W(B_к)$. Отже, й $W(B_i)$ — тавтологія.

Тобто в усіх розглядуваних випадках з індуктивного припущення випливає, що $W(B_i)$ — тавтологія. За принципом індукції для кожного $i \le n$ $W(B_i)$ — тавтологія.

НАСЛІДОК. *Теорія PL є внутрішньо несуперечною теорією.*

Доведення. Припустимо, що існує така формула числення предикатів A^*, що $\vdash A^*$ та $\vdash \neg A^*$. Тоді $W(A^*)$ та $W(\neg A^*)$ — тавтології. Але $W(\neg A^*) = \neg W(A^*)$ згідно з рівністю (4.4.1). Звідси $W(A^*)$ та $\neg W(A^*)$ — тавтології, що неможливо.

У зв'язку з щойно доведеним наслідком виникає питання, чи має місце семантична несуперечність числення предикатів, тобто, чи кожна теорема числення предикатів є формулою, істинною в кожній інтерпретації. Ствердну відповідь на це питання дає наступне твердження.

МЕТАТЕОРЕМА 3. *Кожна теорема числення предикатів є логічно загальнозначущою формулою.*

Доведення. Спочатку впевнимося, що кожний окремий випадок схем аксіом є логічно загальнозначущою формулою. Це вже було встановлено для L1–L3 у [2], а для L4, L5 — у [6]. Тепер покажемо що правила виведення *MP* і В∀ зберігають властивість формул бути істинними в кожній інтерпретації. Для правила *MP* припустимо супротивне. Тоді для деяких формул *A, B* знайдеться хоча б одна інтерпретація і хоча б одне заміщення всіх вільних предметних змінних в ній, при яких формули A та $A \Rightarrow B$ набувають значення "істинність", а формула B — значення "хибність". При цьому істиннісним значенням формули $A \Rightarrow B$ за означенням імплікації є "хибність", що суперечить припущенню. Нарешті, те що правило В∀ зберігає властивість формул бути істинними в кожній інтерпретації, безпосередньо випливає з означення згаданої властивості та з означення істнністного значення формули $\forall xA$. Щоб завершити доведення, досить згадати що кожна теорема числення предикатів є або окремим випадком аксіом L1–L5, або утворюється з них застосуванням скінченого числа раз правил *MP* і *B*∀.

ОЗНАЧЕННЯ 4.4.2. *Теорія першого порядку T називається повною в широкому розумінні, або семантично повною, якщо кожна формула теорії T істинна в інтерпретації, є теоремою T.*

Питання семантичної повноти було розв'язане у 1936 році К. Геделем:

МЕТАТЕОРЕМА 4. (Теорема Геделя про повноту)*. Кожна формула числення предикатів, яка істинна в кожній інтерпретації (лзз), є теоремою числення предикатів.*

Об'єднуючи дві взаємно обернені метатеореми 3 та 4, дістанемо твердження:

МЕТАТЕОРЕМА 5. *Для того, щоб формула числення предикатів була теоремою цього числення, необхідно і достатньо, щоб вона була логічно загальнозначущою.*

Таким чином, у численні предикатів множина теорем, означувана суто синтаксично, і означувана семантично множина логічно загальнозначущих формул збігаються.

Проблема вирішення в численні предикатів зводиться до аналогічної проблеми в логіці предикатів, якщо врахувати факт збігу множини теорем числення предикатів з множиною логічно загальнозначущих формул.

4.5. МАТЕМАТИЧНІ ТЕОРІЇ

Аксіоматичний метод: три стадії розвитку

Аксіоматичний метод — спосіб побудови наукової теорії при якому в її основу покладені деякі вихідні положення, які називають аксіомами, а всі інші положення теорії отримуються як логічні наслідки аксіом за допомогою правил виведення.

Аксіоматичний метод побудови математичних теорій виник у стародавній геометрії. Його відкриття приписують Піфагору (VI ст. до н.е.), але найбільшу популярність цей метод одержав завдяки працям Евкліда (III ст. до н.е.).

Аксіоматичну теорію, подібну до евклідової, у якій значення вихідних термінів пов'язані з визначеними інтуїтивними уявленнями, задаються із самого початку, а логічні засоби виведення теорем з аксіом не фіксуються (діє здоровий глузд, звичайна інтуїтивна логіка), називають *змістовною (неформальною) аксіоматичною теорією (системою)*. Ця змістовна концепція характеризує першу стадію розвитку аксіоматичного методу, що займає великий історичний період від Евкліда (III ст. до н.е.) до Лобачевського (XIX ст.).

Створення М.І. Лобачевским (1826 р.) і Больяй (1832 р.) першої неевклідової геометричної системи ознаменувало собою початок нового періоду розвитку і нової концепції не тільки аксіоматичного методу. Поява різноманітних моделей геометрії Лобачевського-Больяй в об'єктах евклідової геометрії вже означало відхід від концепції змістовної аксіоматичної теорії: у різноманітних моделях однієї і тієї ж аксіоматичної системи значення вихідних термінів різноманітне, хоча логічна структура теорії залишається незмінною. Це і привело до думки про незалежність логічної структури теорії від значень вихідних термінів, а отже, про можливість опису за допомогою однієї теорії цілого класу однорідних структур, що мають однакові властивості, які виражені в аксіомах і теоремах теорії. Ця думка і привела у подальшому Д. Гільберта до нової концепції аксіоматичної теорії.

Висловлюючись сучасною мовою, можна сказати, що гільбертова аксіоматична теорія (на відміну від евклідової) описує не одну конкретну структуру, а клас однорідних структур.

Теорію, подібну до гільбертової, називають тепер *напівформальною* аксіоматичною теорією, тому що в ній логіка, використана в доведеннях теорем, як і в змістовній теорії, не фіксується, не формалізована. Після переліку вихідних понять і аксіом діє знову ж здоровий глузд, інтуїтивна логіка. Саме ж поняття "доведення" залишається на рівні інтуїтивних понять.

Концепція напівформальною аксіоматичної теорії характеризує другу стадію розвитку аксіоматичного методу.

Розвиток аксіоматичного методу привів до постановки таких проблем як несуперечність, повнота, незалежність тієї чи іншої системи аксіом.

Перші результати в цій області приніс метод інтерпретацій, який можна описати таким чином. Нехай кожному вихідному поняттю і відношенню даної аксіоматичної теорії T поставлений у відповідність деякий конкретний математичний об'єкт. Сукупність таких об'єктів M називається полем інтерпретації.

Кожному твердженню U теорії T природно ставиться тепер у відповідність деяке твердження U^* про елементи M, яке може бути істинним або хибним. Тоді відповідно U — істинне або хибне в даній інтерпретації. Поле інтерпретації M та його властивості самі здебільшого є об'єктом, який розглядає деяка, взагалі кажучи, інша, математична теорія Tm, яка також може бути аксіоматичною.

1. Таким чином метод інтерпретацій дозволяє встановити факт відносної несуперечності, тобто доводити твердження типу: "якщо теорія Tm — несуперечлива, то несуперечлива й теорія T"

Нехай теорія T інтерпретована в теорії Tm, таким чином, що всі аксіоми A_i що належать T інтерпретуються як істинні твердження A_i^* з Tm. Тоді кожна теорема теорії T, тобто кожне твердження A логічно виведене із аксіом A_i в T, інтерпретується в Tm деяким твердженням A, виведеним в T_m із інтерпретацій A_i* аксіом A_i, і відповідно істинним.

Останнє твердження спирається на неявно використане припущення про тотожність логічних засобів теорій T та T_m, але практично ця умова здебільшого виконується.

Нехай теорія T — суперечлива, тобто деяке твердження A цієї теорії можна вивести разом з "не A". Тоді із вище викладеного випливає, що твердження A^* та "не A^*" будуть одночасно істинними твердженнями теорії T_m, тобто що T_m — суперечлива.

2. Метод інтерпретацій дає можливість також розв'язувати питання про незалежність системи аксіом. Для доведення того, що аксіома A теорії T не залежить від інших аксіом даної теорії, досить побудувати таку інтерпретацію теорії T, в якій аксіома A була б хибна, а всі інші аксіоми цієї теорії істинні. Іншою формою цього доведення незалежності A є встановлення несуперечливості теорії, яка отримується, якщо в даній теорії T аксіому A замінити її запереченням ($\rceil A$).

Справді, розглянемо систему аксіом $Z = \{A_1, A_2, ..., A_K)$. Аксіома A_i — незалежна, якщо вона не є логічним наслідком інших аксіом Z. Система аксіом Z називається незалежною, якщо будь-яка її аксіома незалежна. Розглянемо систему аксіом $Z' = Z \backslash \{A_i\}$. Аксіома A_i — буде залежною якщо будь-яка інтерпретація системи Z' буде інтерпретацією системи Z.

Нехай $\rceil A_i$ — заперечення аксіоми A_i і розглянемо $Z^* = Z \cup \{\rceil A_i\}$. Будь-яка інтерпретація системи Z^* є також інтерпретацією Z'. Нехай A_i залежить від Z'. Тоді в системі Z^* виконуються аксіоми A_i та $\rceil A_i$, а це значить, що Z^* — суперечлива.

Слабка сторона методу інтерпретації полягає в тому, що в питаннях несуперечності та незалежності системи аксіом він дає можливість одержати результати які носять лише відносний характер.

Проблема несуперечності, найважливіша з проблем, що виникають при побудові аксіоматичної теорії, вирішувалася методом моделей. При цьому несуперечності однієї теорії зводилася до несуперечності іншої і взагалі несуперечність різноманітних математичних теорій зрештою зводилася до несуперечності іншої і взагалі несуперечність різноманітних математичних теорій зрештою зводилася до несуперечності теорії множин. Але в самій теорії множин побудованій на інтуїтивному понятті множини, були виявлені суперечності (парадокси або антиномії). Акліматизація ж теорії множин зажадала доказу її несуперечності, а тут метод моделей вже не можна застосувати, тому що це призводить до порочного кола.

У пошуку виходу з положення, що склалося, Гільберт запропонував (1904 р.) для доведення несуперечності математичних теорій будувати їх у вигляді *формальних систем (формальних теорій),* у яких саме поняття доведення повинно бути точним поняттям, що вивчається спеціальною дисципліною, яку він назвав *теорією доведень,* або *метаматематикою.*

Формальна система будується як точно визначений клас виразів — формул, в якому деяким певним чином виділяється підклас формул, які називаються теоремами. При цьому формули системи безпосередньо не несуть в собі ніякого змісту, і їх можна будувати із довільних, взагалі кажучи, знаків або елементарних символів, керуючись тільки міркуваннями технічної зручності (щоб цей формальний апарат можна було б використовувати для подання яких-небудь теорій). Це і є третя стадія розвитку аксіоматичного методу.

Під математичною теорією будемо розуміти в подальшому формальну аксіоматичну теорію для деякого розділу математики (наприклад, арифметики натуральних чисел, евклідової геометрії, теорії груп і т.п.). Кожна математична теорія будується на базі адекватної формалізованої мови.

Формалізована мова характеризується:

1) індивідними змінними;
2) індивідними сталими;
3) функціональними літерами;
4) предикатними літерами;
5) логічними символами;
6) розділовими знаками.

Вказана множина символів називається *алфавітом теорії.* Скінчена послідовність символів алфавіту теорії називається *словом* або *виразом теорії.*

Виділяється підмножина слів теорії які називаються *формулами теорії* (звичайно вказують ефективну процедуру, що дозволяє за даним виразом з'ясувати, чи є він формулою).

Після того як задана формалізована мова (алфавіт і синтаксис), можна будувати формальну теорію.

З цією метою необхідно задати *логіку* формальної теорії за допомогою систем аксіом і правил виведення, означення формального доведення.

Різні формалізовані мови відрізняються спеціальними аксіомами і своєю *сигнатурою*, тобто множиною предикатних літер, функціональних літер і сталих.

Формалізована мова, визначена таким чином, що квантори навішуються тільки на індивидні змінні, називається *мовою першого порядку*.

Формальна теорія, яка не має жодної спеціальної аксіоми, називається *числення предикатів* (теорія PL).

Природно було б сподіватися, що цей метод формалізації дозволить будувати весь позитивний зміст математичних теорій на такій точній і, здавалось би, надійній основі як поняття формули, що виводиться (теореми формальної системи), а такі принципові питання, як проблема несуперечності математичних теорій, вирішувати у формі доведень відповідних тверджень про формули, що формалізують ці теорії.

Оскільки формальні системи описаного вище типу самі є точними, або, як говорили в школі Гільберта, фінітними математичними об'єктами, то можна було б сподіватися, що вдасться отримати фінітні доведення тверджень про несуперечність. Тобто доведення, які в деякому сенсі були б ефективними, не залежними від таких потужних засобів, як абстракції актуальної нескінченості, які в класичних математичних теоріях як раз і є причиною перешкод в їх обґрунтуванні. Тобто, вимога фінітності засобів, які потрібні для одержання результатів про формальні системи, зокрема теорем про їх несуперечливість, була цілком закономірною особливістю програми Гільберта.

Однак К. Гедель показав (1931 р.), що вже для арифметики принципово неможливо вичерпати весь об'єм її змістовно істинних тверджень класом вивідних формул якої б то не було формальної системи, і що немає ніякої надії отримати якенебудь доведення несуперечності арифметики, оскільки, очевидно, всяке змістовне уточнення поняття фінітного доведення може бути формалізованим у формальній арифметиці.

Все це ставить певні межі можливостям аксіоматичного методу в тому його вигляді, який він мав у гільбертовій формалізації.

4.6. ФОРМАЛЬНА ТЕОРІЯ *Ar* ДЛЯ АРИФМЕТИКИ НАТУРАЛЬНИХ ЧИСЕЛ

Особливу роль серед формальних математичних теорій відіграє формальна арифметика. Оскільки обґрунтування теорії натуральних чисел означає в принципі обґрунтування всієї математики. Справді, доведення несуперечності геометрії Лобачевського зводиться до аналогічного твердження для геометрії Евкліда, обґрунтування останньої зводиться до обґрунтування математичного аналізу, а аналіз — до теорії натуральних чисел.

Мова формальної теорії *Ar* складається з алфавіту числення предикатів до якого приєднано спеціальні символи, а саме:

1) символ бінарного предиката рівності "=";
2) унарна функціональна літера S (безпосереднє слідування за) і дві бінарні функціональні літери "+" і "·";
3) індивідна стала 0.

Отже сигнатурою теорії *Ar* є { 0, S, +, ·, =}.

Поняття терму і формули вводяться так само, як і в теорії PL.

Спеціальні аксіоми

A1 $(x_1 = x_2) \Rightarrow ((x_1 = x_3) \Rightarrow (x_2 = x_3))$;
A2 $\rceil(S(x_1) = 0)$;
A3 $S(x_1) = S(x_2) \Rightarrow x_1 = x_2$;
A4 $x_1 = x_2 \Rightarrow \mathrm{S}(x_1) = \mathrm{S}(x_2)$;
A5 $x_1 + 0 = x_1$;
A6 $x_1 + \mathrm{S}(x_2) = \mathrm{S}(x_1 + x_2)$;
A7 $x_1 \cdot 0 = 0$;
A8 $x_1 \cdot \mathrm{S}(x_2) = x_1 \cdot x_2 + x_1$;
A9 $\mathrm{A}(0) \wedge \forall x_1(\mathrm{A}(x_1) \Rightarrow \mathrm{A}(\mathrm{S}(x_1))) \Rightarrow \forall x_1 \mathrm{A}(x_1)$.

З дев'яти спеціальних аксіом вісім є конкретними аксіомами, а одна (A9) — аксіома індукції — аксіомною схемою, оскільки у її формулюванні є метазнак *A*, який означає довільну формулу цієї теорії.

Правила виведення в теорії A_r, ті ж самі що й у численні предикатів. Разом з ними зберігаються всі похідні правила виведення теорії *PL*, а також до них можна добавити правило індукції, яке ґрунтується на аксіомі (A9):

$$\mathrm{A}(0),\ \forall x_1(\mathrm{A}\ (x_1) \Rightarrow \mathrm{A}(\mathrm{S}(x_1))) \vdash_{Ar} \forall x_1 \mathrm{A}(x_1).$$

1. $\mathrm{A}(0)$	припущення;
2. $\forall x_1(\mathrm{A}(x_1) \Rightarrow \mathrm{A}(\mathrm{S}(x_1)))$	припущення;
3. $\mathrm{A}(0) \wedge \forall x_1(\mathrm{A}(x_1) \Rightarrow \mathrm{A}(\mathrm{S}(x_1)))$	ВвК(1, 2);

4. $A(0) \wedge \forall x_1(A(x_1) \Rightarrow A(S(x_1))) \Rightarrow \forall x_1 A(x_1)$ А9;
5. $\forall x_1 A(x_1)$ MP(3, 4).

Доведемо декілька теорем теорії *Ar* і перш за все леми.

ЛЕМА 1. *Замикання аксіом A1-A8 є теоремами.*

Доведемо цю лему, наприклад для А4, для решти аксіом доведення аналогічні.

1. $x_1 = x_2 \Rightarrow S(x_1) = S(x_2)$ А4;
2. $\forall x_2 (x_1 = x_2 \Rightarrow S(x_1) = S(x_2))$ В∀(1);
3. $\forall x_1 \forall x_2 (x_1 = x_2 \Rightarrow S(x_1) = S(x_2))$ В∀(2).

ЛЕМА 2. *Формули, отримані з аксіом A1-A8 при заміні всіх індивідних змінних довільними термами t є теоремами.*

Доведення обґрунтовується послідовним застосуванням правил В∀, *MP*, L4.

ТЕОРЕМА 1. $t_1 = t_1$.

1. $x_1 = x_2 \Rightarrow (x_1 = x_3 \Rightarrow x_2 = x_3)$ А1;
2. $\forall x_1 \forall x_2 \forall x_3 (x_1 = x_2 \Rightarrow (x_1 = x_2 \Rightarrow x_2 = x_3))$ В∀ 3 рази;
3. $\forall x_1 \forall x_2 \forall x_3 (x_1 = x_2 \Rightarrow (x_1 = x_3 \Rightarrow x_2 = x_3) \Rightarrow$
$(t_1 + 0 = t_1 \Rightarrow (t_1 + 0 = t_1 \Rightarrow t_1 = t_1))$ L4;
4. $t_1 + 0 = t_1 \Rightarrow (t_1 + 0 = t_1 \Rightarrow t_1 = t_1)$ MP(2, 3);
5. $t_1 + 0 = t_1$ А5;
6. $t_1 = t_1$ *MP*(5, 4) 2 рази.

Послідовно доводячи теореми, можна побудувати теорію формальної арифметики *Ar* в якій можна довести всі теореми елементарної арифметики.

Як вже зазначалось, проблема несуперечності значної частини класичної математики може бути зведена до проблеми несуперечності арифметики натуральних чисел.

Доведення несуперечності загострило до себе увагу у 20-х роках. За Гілбертом дане доведення повинно було проводитись надійними фінітними методами. Завдання: довести невивідність у формальній арифметиці певної формули, наприклад $S(0) = 0$.

Дійсно, доведення того, що $\vdash_{Ar} S(0) = 0$ означає внутрішню суперечність теорії А оскільки $\vdash_{Ar} \urcorner(S(0) = 0)$.

1. $\vdash_{Ar} \urcorner(S(x_1) = 0)$ А2;
2. $\vdash_{Ar} \forall x_1 (S(x_1) = 0)$ В (1);
3. $\vdash_{Ar} \forall x_1 (\urcorner((S(x_1) = 0)) \Rightarrow \urcorner(S(0) = 0)$ L4 ;
4. $\vdash_{Ar} \urcorner(S(0) = 0)$ MP(2,3).

Справедливе і обернене твердження: якщо формула $S(0) = 0$ — невивідна в *Ar*, то теорія *Ar* — несуперечна.

Метод геделевої нумерації дає можливість відобразити метаматематику в арифметику. Ідея: оскільки множина всіх символів теорії першого порядку не більш ніж зчисленна, то її можна занумерувати натуральними числами. Припишемо кожному вихідному символу теорії певне натуральне число — $g(c)$ яке назвемо геделевим номером с. Різним символам відповідатимуть різні номери. Формулі теорії першого порядку A яка є скінченною послідовністю вихідних символів $c_1, c_2, ..., c_n$ відповідає геделевий номер

$$g(A) = 2^{g(c_1)} \cdot 3^{g(c_2)} \cdot \ldots \cdot p_n{}^{g(c_n)}.$$

де p_n — n-не просте число. Кожній скінченній послідовності формул теорії відповідатиме геделевий номер. Послідовність $A_1, ..., A_s$ дістане код

$$g(A) = 2^{g(c_1)} \cdot 3^{g(c_2)} \cdot \ldots \cdot p_n{}^{g(c_n)}.$$

За основною теоремою арифметики різні формули і різні послідовності формул одержать відмінні між собою геделеві номери.

Одержана таким чином функція g взаємно-однозначно відображає множину символів, формул і послідовностей формул теорії першого порядку в множину натуральних чисел (можна однозначно встановити за даним геделевим номером, якому саме об'єкту даної формальної теорії він відповідає).

Оскільки доведення формули A в Ar є скінченною послідовністю формул в Ar, останній член якої дорівнює A, то такому доведенню відповідає певний геделів номер.

Завдяки цьому у формальній арифметиці Ar є можливість записати явно замкнену формулу G, яка в звичайній інтерпретації означає змістовно власну невивідність в Ar. Отже отримали доведення існування нерозв'язного в Ar твердження.

ТЕОРЕМА ГЕДЕЛЯ *(про неповноту формальної арифметики). Кожна несуперечна, теорія, що формалізує арифметику натуральних чисел, не є абсолютно повною.*

На закінчення згадаємо про другу важливу теорему метаматематики, яка була доведена в 1936 році американським логіком А. Черчом. В ній стверджується, що *не існує ефективної процедури для розв'язання питання відносно довільної формули формальної теорії, яка містить арифметику натуральних чисел, чи є така формула теоремою теорії чи ні.*

З цієї теореми випливає також і теорема Геделя. Згодом було доведено нерозв'язність великої кількості формальних теорій, зокрема, елементарної теорії груп, елементарної теорії полів.

Розділ 5

ІНТУЇТИВНЕ ПОНЯТТЯ АЛГОРИТМУ

5.1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО АЛГОРИТМИ

Майже у всіх сферах життєдіяльності нас оточують численні та різноманітні алгоритми. Більшість наших дій доведені до несвідомого автоматизму, ми часом і не усвідомлюємо, що вони регламентовані якимось алгоритмом — чіткою системою інструкцій. Автоматизм виконання тих чи інших дій не дозволяє нам усвідомлювати їх алгоритмічну сутність.

Взагалі є багато таких дій, виконуючи які, ми ретельно дотримуємося тієї або іншої інструкції. Це передусім незвичні дії, що професійно нам не властиві.

Чимала кількість алгоритмів зустрічається при вивченні математики у школі. Це, насамперед, алгоритми виконання чотирьох арифметичних дій над різними числами — натуральними, цілими, дробовими, дійсними, комплексними. Ось приклад такого алгоритму: "Щоб від одного десяткового дробу відняти інший, треба: 1) зрівняти число знаків після коми в зменшуваному і від'ємнику; 2) записати від'ємник під зменшуваним так, щоб кома виявилася під комою; 3) зробити вирахування так, як віднімають натуральні числа; 4) поставити в отриманій різниці кому під комами в зменшуваному і від'ємнику".

Застосовуються алгоритми у геометрії: алгоритми геометричних побудов за допомогою циркуля і лінійки (поділ навпіл відрізка і кута, опускання і відновлення перпендикулярів, проведення мимобіжних прямих), алгоритми обчислення площ, об'ємів різних геометричних фігур і тіл.

При вивченні математики в університеті засвоюються процедури обчислення визначників різних порядків, рангів матриць, інтегралів від елементарних функцій, наближених значень коренів рівнянь та систем і т.д. Усі ці процедури є нічим іншим, як алгоритмами.

Загалом, одна з найважливіших задач навчання математики саме й полягає у засвоєнні загальних обчислювальних алгоритмів. Тобто, якщо школяра вчать перемножать стовпцем два числа, то при цьому припускають, що він засвоює не множення конкретних обраних чисел, а універсальний метод (алгоритм), який в подальшому можна застосувати для знаходження добутку довільної пари скінчених чисел

Подібних прикладів можна знайти чимало. Причому не тільки в математиці. Поняття алгоритму стало загальновживаним.

Під *алгоритмом* для деякого класу задач математик розуміє певне загальне правило (чітку систему інструкцій) за допомогою якого розв'язок будь-якої проблеми цього класу може бути знайдено чисто *механічно* та "без жодної винахідливості", якщо, звісно, цей розв'язок існує. Найвідоміший приклад — *алгоритм Евкліда* для знаходження найбільшого спільного дільника двох натуральних чисел або *алгоритм ділення*. Алгоритм Евкліда після скінченного числа кроків завжди приводить до деякого результату: він *обривається*. Навпаки, алгоритм ділення обривається тільки в тих випадках, коли частка, що ним визначається, має скінченне десяткове представлення, і тільки в цих випадках він приводить до певного результату. Алгоритми широко поширені як у практиці, так і в науці, і вимагають більш уважного до себе відношення і ретельного вивчення математичними методами.

Інтерес математиків до алгоритмів досить великий, оскільки алгоритми дозволяють отримати — принаймні принципово — схематичний розв'язок певного класу задач і тим самим — принаймні принципово — тривіалізувати певну область математики.

Слово "алгоритм" походить від імені арабського математика Мухаммад ібн Муса аль-Хорезми (787 — біля 850). В його трактаті, написаному арабською, латинська версія якого відноситься до XII століття і починається словами «Dixit algorizm», тобто «Сказав аль-Хорезми», була описана індійська позиційна система чисел і сформульовані правила виконання чотирьох арифметичних дій над числами в десятковому запису. Цим він вніс значний вклад у розповсюдження методів обчислень, що тоді існували. Прогрес у розвитку таких методів породив доживше до наших днів уявлення про те, що кінцевий розв'язок будь-якої поставленої математичної, навіть філософської проблеми повинен бути її алгоритмічним розв'язком.

Серед представників цього погляду згадаємо Декарта, Лейбніца та Гільберта. Декарт розбудовував аналітичну геометрію з наміром зробити геометрію доступною алгебраїчним методам обчислень і тим самим істотно просунутись на шляху її алгоритмізації. Лейбніц протягом усієї своєї творчої діяльності займався уточненням та розв'язком алгоритмічних проблем. Йому ж належить перша спроба придумати пристосовану для цієї мети *автоматично працюючу машину*; проте ця спроба не була успішною. Гільберт винятково сильно стимулював такі дослідження, особливо своїми закликами до алгоритмічного розв'язку відомих класів задач (десята проблема Гілберта, проблема розв'язності для логіки предикатів).

Віра в універсальність алгоритмічних методів була підірвана роботою Гьоделя, який вперше довів *алгоритмічну нерозв'язність* деяких математичних проблем. Точніше, було показано, що відомі математичні проблеми не можна розв'язати за допомогою алгоритмів з деякого точно визначеного класу алгоритмів. Значення результату Гьоделя залежить при цьому від ступеня співпадання цього алгоритмічного класу та класу всіх алгоритмів у інтуїтивному сенсі. Тим самим виникла цілком нова ситуація. До тих пір, поки ми вірили у можливість того, що *всі* поставлені математичні задачі можна алгоритмічно розв'язати, в нас не було підстав уточнювати поняття алгоритму: коли для розв'язку деякого класу проблем пропонувався конкретний алгоритм, виникала угода вважати цей алгоритм дійсно алгоритмом. Тільки твердження про алгоритмічну нерозв'язність, тобто *доведення неможливості*, в якому містилось би висловлення про всі алгоритми, вимагає попереднього уточнення.

Починаючи з 1935 року було запропоновано низку уточнень поняття алгоритму. Сьогодні, за невеликим виключеннями, панує переконання, що ці поняття є адекватним виразом інтуїтивного представлення.

Перед тим, як перейти до математичного вивчення поняття алгоритму, спробуємо уважно проаналізувати приклади алгоритмів, виявити їх загальні властивості та особливості.

1. Кожен алгоритм припускає наявність деяких *початкових* або *вхідних даних*, а його застосування, в наслідку, приводить до одержання визначеного шуканого *результату*.

2. Застосування кожного алгоритму здійснюється шляхом виконання послідовності деяких елементарних дій, ці дії називають *кроками*. Причому, виконання наступного кроку можливе тільки після завершення всіх операцій на попередньому кроці, а процес їх виконання називають *алгоритмічним процесом*. Так визначається властивість *дискретності алгоритму*.

3. Обов'язковою умовою, якій задовольняє алгоритм, є його *визначеність*. Це значить, що вимоги (правила) алгоритму з одинаковим успіхом можуть бути виконані будь-якою іншою людиною й у будь-який час, причому результат вийде той же самий. Іншими словами, правила алгоритму настільки точні, виразні, що не допускають ніяких двозначних тлумачень і ніякої сваволі зі сторони виконавця. Вони єдиним і цілком певним шляхом усякий раз приводять до шуканого результату. *Визначеність (детермінованість) алгоритму* полягає в тому, що сукупність проміжних величин на кожному кроці однозначно визначається системою величин, наявних на попередньому кроці. Дана влас-

тивість означає, що результат виконання алгоритму не залежить від того хто (чи що) його виконує, а визначається тільки вхідними даними та кроками самого алгоритму. Це наводить на думку, що виконання тих або інших алгоритмів може бути доручено машині, що широко й робиться на практиці.

4. *Елементарність кроків* — закон отримання наступної сукупності величин з попередньої має бути простим та локальним. Яку дію (крок) можна вважати елементарною, визначається особливостями виконувача алгоритму.

5. *Загальний характер* алгоритму є його істотною рисою, тобто можливість застосовувати його до великого класу початкових даних, можливість досить широко ці початкові дані видозмінювати. Іншими словами, кожен алгоритм покликаний вирішити ту або іншу *загальну проблему*, тобто розв'язувати клас однотипових задач.

Наприклад, задача знаходження найбільшого спільного дільника чисел 4 та 6 є одинична проблема (можна розв'язати її і без застосування алгоритму Евкліда), але задача находження найбільшого спільного дільника довільних натуральних чисел m та n — вже проблема загальна. Суть алгоритму Евкліда полягає в тому, що він приводить до бажаного результату поза залежністю від вибору конкретної пари натуральних чисел, у той час як при розв'язанні зазначеної одиничної проблеми можна запропонувати такий спосіб, що виявиться непридатним для іншої пари натуральних чисел.

6. Говорячи про початкові дані для алгоритму, мають на увазі так звані *допустимі початкові дані,* тобто такі початкові дані, що сформульовані в термінах даного алгоритму. Серед допустимих початкових даних для алгоритму можуть бути такі, до яких він застосовний, тобто рушаючи від яких можна одержати шуканий результат, а можуть бути такі, до яких даний алгоритм не застосуємо, тобто відправляючись від яких шуканого результату одержати не можна. Незастосовність алгоритму до можливих початкових даних може полягати або в тому, що алгоритмічний процес ніколи не закінчується (у цьому випадку говорять, що він нескінченний), або в тім, що його виконання під час одного з кроків наштовхується на перешкоду, заходить у безвихідь (у цьому випадку говорять, що він безрезультатно обривається).

Отже, підводячи підсумки наведених характерних властивостей і особливостей алгоритму, можемо сформулювати таке описове інтуїтивне означення цього поняття.

Під *алгоритмом* розуміється чітка система інструкцій, що визначає дискретний детермінований процес, який веде від змінних початкових даних (входів) до шуканого результату (виходу), якщо такий існує, через скінченне число кроків роботи алгоритму;

якщо ж шуканого результату не існує, то обчислювальний процес або ніколи не закінчується, або потрапляє в безвихідь.

Поняття алгоритму, що визначається переліком властивостей 1-6, також неможливо вважати строгим, оскільки формулювання властивостей містять терміни «спосіб», «простий», «локальний» та інші, точний сенс яких не встановлено. У подальшому таке означення будемо називати *нестрогим (інтуїтивним)* поняттям алгоритму.

5.2. НЕОБХІДНІСТЬ УТОЧНЕННЯ ПОНЯТТЯ АЛГОРИТМУ

Хоча поняття алгоритму формувалося з якнайдавніших часів, але до кінця першої третини XX століття математиків задовольняло інтуїтивне поняття цього об'єкту. Термін алгоритм вживався в математиці лише у зв'язку з тими або іншими конкретними алгоритмами. Твердження про існування алгоритму для розв’язку завдань даного типу супроводжувалося фактичним його описом.

Виникає питання: чи так вже й важливо та необхідно мати точне означення алгоритму, якщо і без нього можливе складання та застосування алгоритмів. Проте були сформульовані такі проблеми, алгоритмічне розв’язання яких було неочевидне. Справді, для доведення існування алгоритму необхідно просто розв’язати задачу, користуючись набором відомих засобів, або в разі їх відсутності запропонувати нові засоби, тоді достатньо інтуїтивного поняття алгоритму, щоб переконатись, що цей процес є алгоритмом. Важче довести факт неможливості побудови алгоритму розв’язку деякої задачі (або класу задач), без точного означення алгоритму ця проблема втрачає сенс. Для точного доведення відсутності якогось об'єкту необхідно мати його точне математичне означення. Абсолютно аналогічна ситуація склалася свого часу в математиці, коли назріла необхідність уточнення таких понять, як неперервність, крива, поверхня, довжина, площа, об'єм тощо.

Парадокси, виявлені в основах математики на початку минулого століття, призвели до появи різних концепцій та течій, покликаних ці парадокси усунути. У 20-і роки постали питання про те, що ж таке строга вивідність та ефективне обчислення. Поняття алгоритму саме повинно було стати об'єктом математичного дослідження і тому потребувало строгого означення. Крім того, до цього призводив розвиток фізики та техніки, що швидко наближав початок епохи електронно-обчислювальних машин.

Іншою вимогою побудови точного означення алгоритму була невизначеність поняття «елементарності кроку» при виконанні алгоритмічних дій. Допоки математика вивчала числові об'єкти, то елементарними вважались арифметичні операції, а також декілька логічних операцій. Проте після початку дослідження складних об'єктів — векторів, матриць, функцій, поняття елементарності кроку алгоритму перестало бути очевидним. Наприклад, чи можна вважати елементарним кроком находження оберненої матриці або інтеграла?

У випадку, коли задача допускає побудову декількох алгоритмів розв'язку, то з практичної та з теоретичної сторони виявляється істотним питання їх співставлення та вибору найбільш ефективного, що, зрозуміло, неможливе без строгого означення алгоритму.

Отже, виникла потреба у точному визначенню поняття алгоритму, тобто у максимально загальному означенні, яке охоплювало би всі можливі види алгоритмів. Спроби сформулювати таке поняття привели до появи теорії алгоритмів як самостійної науки, котра разом з математичною логікою вивчає основні засоби математики — методи доведень, властивості математичних процедур, способи побудови аксіоматичних теорій та інше.

Уточнення поняття алгоритму проводиться в рамках алгоритмічних моделей. Модель визначає набір засобів, використання яких допустиме при розв'язанні задачі. Розрізняють *теоретичні* та *практичні* алгоритмічні моделі. З теоретичної точки зору моделі повинні бути *універсальними*, тобто дозволяють описати будь-який алгоритм, а з іншого боку — максимально *простими*, тобто використовують мінімум засобів для розв'язку задачі. У практичних і прикладних моделях більш вагомим є *зручність* програмування та *ефективність* обчислень, тому їх засоби більш різноманітні та складніші, що обтяжує теоретичний аналіз. Перші роботи по уточненню поняття алгоритму та його вивченню, тобто з теорії алгоритмів, були виконані в 1936-1937 роках математиками А. Т'юрингом, Э. Постом, Ж. Ербраном, К. Геделем, А.А. Марковим, А. Чорчем. Було вироблено декілька означень поняття алгоритму, але згодом з'ясувалося, що всі вони рівносильні між собою, тобто визначають одне і те ж поняття.

При побудові строгого означення поняття алгоритму в теоретичному підході історично виділились три основні напрямки.

Перший напрямок пов'язує поняття алгоритму з традиційними уявленнями — процедурами, що дозволяють вирахувати значення числових функцій, які залежать від цілочисельних значень аргументів. Такі функції отримали назву *обчислюваних*. Поняття обчислюваної функції не є строгим, як і поняття алгоритму. Проте, завдяки роботам А. Чорча, К. Геделя, С. Кліні, була обґрунтована тотожність класу всюди визначених обчислюваних функцій з класом частково рекурсивних функцій, який визначається строго. Це дозволило звести проблему алгоритмічної розв'язності до доведення можливості (або неможливості) побудови рекурсивної функції, що розв'язує задачу. Саме так А. Чорчу вдалось довести нерозв'язність однієї з проблем математичної логіки — обчислювання істинності предикатів.

Другий напрямок пов'язаний з машинною тематикою. Тут алгоритм трактується як детермінований пристрій, що здатен виконувати лише точно фіксовану множину операцій. Тобто алгоритмічні процеси, це процеси, які може здійснювати відповідним чином налаштована машина. За цією ідеєю в точних математичних термінах були описані відповідні класи машин, та було доведено, що цими машинами можна здійснити або зімітувати всі алгоритмічні процеси. Цей підхід було розвинено в роботах Е. Поста, А. Т'юринга. Доведення алгоритмічної розв'язності задачі зводиться до доведення існування машини, що її розв'язує.

Третій напрямок алгоритмічних моделей — це перетворення слів у довільних алфавітах, в яких операціями є заміни шматків слів іншим словом. Основною теоретичною моделлю цього типу є нормальні алгоритми Маркова.

Теорія алгоритмів мала істотний вплив на розвиток обчислювальних машин і практику програмування. В теорії алгоритмів були передбачені основні концепції, що закладені в апаратуру та мови програмування. Згадувані вище головні алгоритмічні моделі математично еквівалентні, проте на практиці вони істотно розрізняються ефектами складності, що виникають при реалізації алгоритмів, та породили різномаїття напрямків у програмуванні. Зокрема, мікропрограмування будується на ідеях машини Т'юринга, структурне програмування взяло свої конструкції з теорії рекурсивних функцій, мови символьної обробки інформації беруть початок від нормальних алгоритмів Маркова та систем Поста.

Зараз алгоритмічні концепції відіграють в розумінні процесів автоматичної обробки інформації фундаментальну роль, аналогічну ролі писемності.

Ще слід відмітити те, що строго формалізований підхід при означенні поняття «алгоритм» використовується лише безпосередньо в самій математичній теорії алгоритмів, де досліджуються загальні властивості алгоритмів. У практичних застосуваннях, в тому числі в інформатиці, строга формалізація може привести до значного ускладнення задачі, до того ж, можна вказати ряд ситуацій, в яких допустимо відходження від неї.

Вправи

1. Скласти алгоритм знаходження кореня квадратного з числа.
2. Скласти алгоритм знаходження найбільшого спільного дільника.
3. Скласти алгоритм знаходження найменшого спільного кратного.
4. Скласти алгоритм множення матриць.
5. Скласти алгоритм знаходження оберненої матриці.
6. Скласти алгоритм знаходження порожньої кулі за 2 зважування на терезах, якщо відомо, що серед 8 однакових куль є одна порожня.
7. Скласти алгоритм пошуку найбільшого та найменшого чисел з множини $M = \{a_1, a_2, ..., a_n\}$.
8. Скласти алгоритм пошуку найбільшого та найменшого добутку чисел з множини $M = \{a_1, a_2, ..., a_n\}$.
9. Скласти алгоритм ходу пішака в шахах.
10. Скласти алгоритм ходу коня в шахах.

Розділ 6

РЕКУРСИВНІ ФУНКЦІЇ

6.1. ОБЧИСЛЮВАНІ ФУНКЦІЇ. РОЗВ'ЯЗУВАНІ МНОЖИНИ. ПЕРЕРАХОВНІ МНОЖИНИ

Нехай маємо дві множини X та Y.

Якщо деяким елементам множини X поставлено у відповідність однозначно визначені елементи множини Y, то говорять, що задано *часткову функцію* з X в Y.

Сукупність тих елементів множини X, у яких є відповідні елементи множини Y, називається *областю визначення функції,* а сукупність тих елементів множини Y, які відповідають елементам множини X, називається *множиною значень функції.*

Якщо область визначення функції з X в Y співпадає з множиною X, то функція називається *всюди визначеною.*

Вихідна ідея точного означення алгоритму, що опирається на поняття рекурсивної функції, полягає в тому, що будь які дані (в т.ч. дискретні) можна закодувати натуральними числами в деякий системі числення, і тоді кожне їх перетворення зведеться до послідовності обчислювальних операцій, а результат обрахунків також буде цілим числом. У цьому підході кожен алгоритм, єдиний для даної числової функції, обчислює її значення, а його елементарними кроками є звичайні арифметичні та логічні операції. Такі функції отримали назву *обчислюваних.*

Розглянемо функції f від одного або декількох аргументів, задані на множині $N = \{1, 2, 3, 4, ..., n, ...\}$ всіх натуральних чисел або на деяких її підмножинах (часткові функції) і такі, що приймають значення в множині N. Отже, розглядаються функції f, що є відображеннями декартового n-ого степеня N^n в N. Область визначення Df функції f є підмножина множини $N^n = N \times ... \times N$:

$$\boldsymbol{D}f = \{(x_1, ..., x_n) : f(x_1, ..., x_n) \text{ — визначено}\}.$$

Область зміни (значень) f є підмножина множини N:

$$\boldsymbol{I}f = \{y \in N : (\exists x_1, ..., \exists x_n) \wedge (f(x_1, ..., x_n) = y)\}.$$

ОЗНАЧЕННЯ 6.1.1. *Функція $f(x_1, x_2, ..., x_n)$ називається обчислюваною, якщо існує алгоритм, що дозволяє обчислювати її значення для тих наборів аргументів, для яких вона визначена, і який працює нескінченно, якщо функція для даного набору значень аргументів не визначена.*

Оскільки поняття алгоритму в цьому означенні береться інтуїтивним, то й поняття обчислюваної функції також інтуїтивне. Втім, при переході від алгоритмів до обчислюваних функцій виникає одна дуже істотна обставина. Сукупність процесів, що задовольняють умовам 1-6 з п. 1.1 а, отже, таких, що підлягають під інтуїтивне поняття алгоритму, доволі велика. Проте, сукупність обчислюваних функцій для різномаїття поняття процесів, що задовольняють вказаним умовам, виявилась однією й тією ж, до того ж її легко описувати у звичайних математичних термінах. Ця точно описана сукупність числових функцій, що співпадає з сукупністю всіх обчислюваних функцій за самого широкого до тепер відомого розуміння алгоритму, носить назву сукупності *рекурсивних функцій*. До розгляду яких перейдемо згодом.

Нехай $M \subseteq N^n = N \times N \times \ldots \times N$.

ОЗНАЧЕННЯ 6.1.2. *Множина М називається розв'язуваною, якщо є алгоритм для з'ясування того, належить або не належить довільний елемент до цієї множини.*

Відоме поняття *характеристичної функції* множини M. Це функція χ_M, що задана на множині M, та приймає значення на двоелементній множині$\{0, 1\}$, означена так:

$$\chi_M(x_1,\ldots,x_n) = \begin{cases} 0, \text{ якщо } (x_1,\ldots,x_n) \notin M; \\ 1, \text{ якщо } (x_1,\ldots,x_n) \in M. \end{cases}$$

Зрозуміло, що множина M розв'язувана тоді і тільки тоді, коли її характеристична функція χ_M обчислювана.

Нехай тепер $M \subseteq N$.

ОЗНАЧЕННЯ 6.1.3. *Множина $M \subseteq N$ називається (рекурсивно, або ефективно, або алгоритмічно) перераховною, якщо М або порожня, або є областю значень деякої обчислюваної функції, або, іншими словами, якщо існує алгоритм для послідовного породження (переліку) всіх її елементів.*

ТЕОРЕМА 6.1.1. *Якщо множини М і L перераховні, то перераховні множини $M \cup L$ та $M \cap L$.*

Доведення. Якщо є алгоритми для породження елементів множин M та L, то алгоритм для породження множини $M \cup L$ отримаємо шляхом їх одночасного застосування. Отже, множина $M \cup L$ перераховується.

Нехай тепер алгоритм Ω_M послідовно породжує елементи $m_1, m_2, m_3, \ldots$ множини M, а алгоритм Ω_L послідовно породжує елементи $l_1, l_2, l_3., \ldots$ множини L. Тоді алгоритм для переліку елементів множини $M \cap L$ такий.

По черзі за допомогою алгоритмів Ω_M і Ω_L породжуються елементи $m_1, l_1, m_2, l_2, m_3, l_3$, і т.д. Кожен знову породжений елемент m_i, порівнюється зі всіма раніше породженими елементами l_i. Якщо m_i співпадає з одним з них, то його зараховують до множини $M \cap L$. Інакше треба переходити до породження елементу l_i, і порівнювати його зі всіма раніше породженими елементами m_i і т.д. Описана процедура дозволяє ефективно перераховувати всі елементи множини $M \cap L$, що й потрібно було довести.

ТЕОРЕМА 6.1.2. *Нехай $M \subseteq N$. Множина M розв'язувана тоді і тільки тоді, коли вона сама та її доповнення **C**M перераховні.*

Доведення. Необхідність. Нехай M — розв'язувана множина. Можемо вважати, що $M \neq \varnothing$, тобто є елемент $a \in M$. Тоді характеристична функція χ_M множини M, як було зауважено вище, обчислювана, тобто є алгоритм для її обчислення.

Побудуємо алгоритм для переліку множини M. Розглянемо функцію

$$f(x) = \begin{cases} x, \text{ якщо } \chi_M = 1; \\ a, \text{ якщо } \chi_M = 0. \end{cases}$$

Звідси $M = \{f(0), f(1), f(2), f(3), \ldots\}$, тобто M є множина значень функції f, яка, напевно, обчислювана зважаючи на обчислюваність функції $\chi_M(x)$. Отже, множина M перераховується.

Аналогічно, обравши елемент $b \notin M$, будуємо функцію:

$$g(x) = \begin{cases} b, \text{ якщо } \chi_M = 1, \\ x, \text{ якщо } \chi_M = 0, \end{cases}$$

яка також обчислювана, зважаючи на обчислюваність функції $\chi_M(x)$. Крім того, $M = \{g(0), g(1), g(2), g(3),\ldots\}$, тобто **C**$M$ є множина значень обчислюваної функції g. Отже, множина **C**M також перераховна.

Достатність. Нехай множини M та **C**M перераховні, тобто $M = \{f(0), f(1), f(2), \ldots\}$ та **C**$M = \{g(0), g(1), g(2), \ldots\}$, де f та g — деякі обчислювані функції. Тоді алгоритм, що з'ясовує, належить чи ні довільне число n множині M, діє так. Послідовно породжуємо елементи $f(0), g(0), f(1), g(1), f(2), g(2), \ldots$ та на кожному кроці отриманий елемент порівнюємо з n. Оскільки n повинно належати або M, або **C**M, то на останньому кроці одержимо $f(k) = n$ або $g(k) = n$. Якщо $f(k) = n$, то $n \in M$, а якщо $g(k) = n$, то $n \in$ **C**M, тому, $n \notin M$. Отже, множина M розв'язна. Теорему доведено.

Зауважимо, що існує ефективний перелік всіх впорядкованих пар натуральних чисел, який називається *діагональним методом* або *діагональним процесом*:

$$\begin{array}{llll}
(0,0) & (0,1) & (0,2) & (0,3)\ \dots \\
(1,0) & (1,1) & (1,2) & (1,3)\ \dots \\
(2,0) & (2,1) & (2,2) & (2,3)\ \dots \\
(3,0) & (3,1) & (3,2) & (3,3)\ \dots \\
\dots & \dots & \dots & \dots
\end{array}$$

Перелік здійснюється послідовним проходженням по діагоналях, починаючи з лівого верхнього кута. Першими елементами цього переліку є: (0,0); (1,0), (0,1); (2,0), (1,1), (0,2); (3,0),.... Можна перевірити, що в даній послідовності пара (m, n) стоїть на місці з номером

$$C(m,n) = \frac{m^2 + 2mn + n^2 + 3m + n + 2}{2}.$$

Теорема 6.1.3. *Існує множина натуральних чисел, яка перераховна, але не розв'язувана.*

Доведення. За попередньою теоремою досить навести приклад такої множини K натуральних чисел, яка перераховується, а її доповнення $\mathbf{C}K$ не перераховне.

Нехай $M_0, M_1, M_2, \dots$ — ефективний перелік всіх множин, що перераховуються, тобто такий перелік, що за будь-яким заданим числом p можна відновити саму множину M_p. (Можливість такого переліку з'явиться після того, як буде уточнено та дано строге означення поняття алгоритму).

Розглянемо тепер алгоритм $\mathfrak{A}$, який послідовно породжує всі елементи такої множині K. На кроці з номером $C(m, n)$ цей алгоритм обчислює m-ий елемент множині M_n, і якщо елемент співпадає з n, то він відносить його до множини K. Тому $n \in K \Leftrightarrow n \in M_n$.

Отже, K породжується алгоритмом $\mathfrak{A}$, тобто K перераховна. Оскільки доповнення $\mathbf{C}K$ множини K складається зі всіх таких n, що $n \notin M_n$, то K відрізняється хоч одним елементом від будь-якої множини, що є перераховною. Тому K не перераховується. Отже, за попередньою теоремою 6.1.2, множина K не розв'язна, що і завершує доведення.

6.2. РЕКУРСИВНІ ФУНКЦІЇ ТА ЇХ ВЛАСТИВОСТІ

Поняття рекурсивної функції є одним з математичних уточнень інтуїтивного поняття обчислюваної функції. Розглянемо основні поняття цієї алгоритмічної теорії — теорії рекурсивних функцій.

Клас рекурсивних функцій будується подібно тому, як будувалася аксіоматична теорія висловлень. Вибираються деякі первинні, в якомусь сенсі найпростіші, функції (в численні висловлень вибиралися первинні формули — аксіоми). Потім формулюються деякі правила (в численні висловлень вони називалися правилами виведення), на основі яких можна будувати нові функції з тих, що вже є. Назвемо такі правила операторами. Тоді необхідним класом функцій буде сукупність всіх функцій, які виходять з найпростіших за допомогою вибраних операторів.

Побудуємо клас рекурсивних функцій відповідно до заявлених принципів. Нагадаємо, що розглядаються функції, задані на множині натуральних чисел та такі, що приймають натуральні значення. Функції беремо часткові, тобто визначені, взагалі кажучи, не для всіх значень аргументів. Як початкові, найпростіші, функції виберемо такі:

$S(x) = x + 1$ *(функція слідування);*

$O^n(x_1, x_2, \ldots, x_n) = 0$ *(нуль-функція);*

$I^n_m(x_1, x_2, \ldots, x_n) = x_m$ *(функції-проектори,* $1 \le m \le n$*).*

Як відомо кожна алгоритмічна модель, в тому числі й рекурсивні функції, повинна передбачати визначення елементарних кроків алгоритму та способів побудови з них певної послідовності перетворень, що забезпечують потрібну послідовність обробки даних. В рекурсивній моделі такими «елементарними кроками» є ці найпростіші числові функції $S(x)$, O^n, I^n_m комбінацією яких будуються всі більш складні. Вказані найпростіші функції всюди визначені та інтуїтивно обчислювані.

Виберемо оператори, за допомогою яких будуватимемо нові функції. Це такі три оператори: суперпозиції, примітивної рекурсії та мінімізації.

Оператор суперпозиції. Говоритимемо, що n-місна функція ψ одержана з m-місної функції φ та n-місних функцій $f_1, \ldots, f_m$ за допомогою оператора суперпозиції, якщо для всіх $x_1, \ldots, x_n$, справджується рівність:

$$\psi(x_1, \ldots, x_n) = \varphi(f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)).$$

Якщо ми вміємо обчислювати функції $\varphi, f_1, f_2, ..., f_n$, то функція ψ, також може бути обчислена. Зрозуміло також, що якщо всі функції $\varphi, f_1, f_2, ..., f_n$ всюди визначені, то і функція ψ також всюди визначена. Отже, якщо функції $\varphi, f_1, f_2, ..., f_n$ інтуїтивно обчислювані, то буде інтуїтивно обчислюваною й функція ψ.

Приклад 6.2.1. Обчислити значення $\varphi^2(S(x), 0^1)$.

Для цього значення простішої функції 0^1 треба підставити в $S(x) = x + 1$. Але $0^1(x) = 0$, отже $\psi(x) = S(0^1) = 0 + 1 = 1$.

Приклад 6.2.2. Обчислити значення $\varphi^3(I_2^2, I_1^1, 0^1)$.

В цьому випадку $\psi = I_2^2(I_1^1, 0^1) = I_2^2(x_1, 0) = 0$.

Оскільки поняття суперпозиції функцій добре відоме, то не зупинятимемося на ньому докладніше.

Оператор примітивної рекурсії. Скажемо що $(n+1)$-місна функція φ отримана з n-місної функції f та $(n+2)$-місної функції g за допомогою оператора примітивної рекурсії, якщо для будь-яких значень $x_1, ..., x_n, y$ виконується рівність

$$\varphi(x_1, ..., x_n, 0) = f(x_1, ..., x_n),$$
$$\varphi(x_1, ..., x_n, 1) = g(x_1, ..., x_n, 0, \varphi(x_1, ..., x_n, 0)), \quad (6.2.1)$$
$$......$$
$$\varphi(x_1, ..., x_n, y+1) = g(x_1, ..., x_n, y, \varphi(x_1, ..., x_n, y)).$$

Ці рівності називають *схемою примітивної рекурсії.*

Символічно примітивна рекурсія позначається $\varphi = R(f, g)$. в цьому запису R розглядається як символ двомісної часткової операції, що визначена на множині всіх часткових функцій. Із співвідношення (6.2.1) випливає, зокрема, що якщо f, g є всюди визначеними, то й φ також є всюди визначеною. З (6.2.1) маємо, що якщо ми вміємо знаходити значення функцій f та g, то значення функції $\varphi(x_1, ..., x_n, y+1)$ можна обчислювати «механічно», знаходячи значення послідовно на попередніх кроках.

Введемо означення. Часткова функція $f(x_1, ..., x_n)$ називається *примітивно рекурсивною*, якщо її можна отримати скінченим числом операцій суперпозиції та примітивної рекурсії, виходячи лише з найпростіших функцій $S(x)$, O^n, I_m^n.

Якщо операції суперпозиції та примітивної рекурсії застосувати до всюди визначених функцій, то в результаті отримаємо також всюди визначену функцію. Зокрема, всі примітивно рекурсивні функції всюди визначені.

Приклад 6.2.3. Покажемо, що функцію $s(x, y) = x + y$ можна одержати з найпростіших за допомогою оператора примітивної рекурсії. Для функції справджуються тотожності

$$x + 0 = x, x + (y + 1) = (x + y) + 1,$$

які можна записати у вигляді

$$s(x, 0) = x,\ s(x, y+1) = s(x, y) + 1.$$

або

$$s(x, 0) = I_1^1(x),\ s(x, y+1) = s(s(x, y)).$$

Це і є схема рекурсії, що ґрунтується на функціях I_1^1 та S.

Приклад 6.2.4. Покажемо, як, виходячи з найпростіших функцій, за допомогою оператора примітивної рекурсії можна одержати нову функцію, яку називають *зрізаною різницею*:

$$x \div y = \begin{cases} x - y, & \text{якщо} \quad x \geq y, \\ 0, & \text{якщо} \quad x < y. \end{cases}$$

По-перше, відзначимо, що функція $x \div 1$, яку отримаємо з функції $x \div y$ фіксацією другого аргументу, задовольняє таким співвідношенням:

$$0 \div 1 = 0 = O(x),\ (x + 1) \div 1 = x = I_1^2(x, y),$$

тобто одержана з найпростіших функцій $O(x)$ і $I_1^2(x, y)$ за допомогою оператора примітивної рекурсії.

По-друге, легко зрозуміти, виходячи з означення зрізаної різниці, що ця функція задовольняє також рівності

$$x \div 0 = x = I_1^2(x, y),$$

$$x \div (y + 1) = (x \div y) \div 1 = h(x, y, x \div y)$$

для будь-яких x та y. Тотожність показує, що двомісна функція $x \div y$ одержана за допомогою оператора примітивної рекурсії з найпростішої функції $I_1^2(x, y)$ та функції $h\ (x, y, z) = z \div 1$.

Приклад 6.2.5. Зрозуміле співвідношення для операції множення, що вводиться аналогічно операції додавання:

$$p(x,0) = x \cdot 0 = 0 = O(x),\ p(x, y + 1) = x \cdot y + x =$$

$$= p(x, y) + x = x + p(x, y) = s(x, p(x, y)).$$

Це свідчить, що функція $p(x, y) = x \cdot y$ одержана з найпростішої функції $O(x)$ та функції $s(x, y) = x + y$ за допомогою оператора примітивної рекурсії.

Введемо третій, заключний, оператор.

Оператор мінімізації. Говоритимемо, що n-місна функція φ отримана з $(n + 1)$-місних функцій f_1, і f_2 за допомогою оператора мінімізації, або оператора найменшого числа, якщо для будь-яких $x_1, ..., x_n, y$ рівність $\varphi(x_1, ..., x_n) = y$ виконується тоді і тільки тоді, коли значення $f_i(x_1, ..., x_n, 0), ..., f_i(x_1, ..., x_n, y - 1)$ $(i = 1, 2)$ визначені та попарно нерівні:

$$f_1(x_1, ..., x_n, 0) \neq f_2(x_1, ..., x_n, 0),$$

$$\dots\dots\dots\dots$$

$$f_1(x_1, ..., x_n, y - 1) \neq f_2(x_1, ..., x_n, y - 1),$$

а

$$f_1(x_1, ..., x_n, y) = f_2(x_1, ..., x_n, y).$$

Стисло кажучи, величина $\varphi(x_1, ..., x_n)$ дорівнює найменшому значенню аргументу y, при якому виконується остання рівність. Використовується таке позначення:

$$\varphi(x_1, ..., x_n) = \mu y[f_1(x_1, ..., x_n, y) = f_2(x_1, ..., x_n, y)].$$

В окремому випадку може бути $f_2 = 0$. Тоді

$$\varphi(x_1, ..., x_n) = \mu y[f(x_1, ..., x_n, y) = 0].$$

Оператор мінімізації називається також *μ-оператором.*

Приклад 6.2.6. Дослідимо функцію, яку отримаємо за допомогою оператора мінімізації:

$$d(x, y) = \mu z[y + z = x] = \mu z[s(I_2^3(x, y, z), I_3^3(x, y, z)) = I_1^3(x, y, z)].$$

Обчислимо, наприклад, $d(6, 2)$. Для цього потрібно покласти $y = 2$ і, надаючи змінній z послідовно значень 0, 1, 2, ..., кожного разу обчислювати суму $y + z$. Як тільки вона стане рівною 6, то відповідне значення z прийняти за значення $d(6, 2)$. Обчислюємо:

$$\begin{array}{ll} z = 0, & 2 + 0 \neq 6; \\ z = 1, & 2 + 1 \neq 6; \\ z = 2, & 2 + 2 \neq 6; \\ z = 3, & 2 + 3 \neq 6; \\ z = 4, & 2 + 4 = 6. \end{array}$$

Отже, $d(6, 2) = 4$.

Спробуємо обчислити за цим правилом d(3, 4):

$$\begin{array}{ll} z = 0, & 4 + 0 = 4 > 3; \\ z = 1, & 4 + 1 = 5 > 3; \\ z = 2, & 4 + 2 = 6 > 3; \\ \ldots\ldots & \end{array}$$

Бачимо, що даний процес продовжуватиметься нескінченно. Отже, $d(3, 4)$ не визначене.

Отже, маємо, $d(x, y) = x - y$.

Приклад 6.2.7. Аналогічно, за допомогою оператора мінімізації можна одержати часткову функцію, що визначає частку від ділення двох натуральних чисел:

$$x/y = q(x, y) = \mu z[yz = x] =$$
$$= \mu z[p(I_2^3(x, y, z), I_3^3(x, y, z)) = I_1^3(x, y, z)].$$

Зрозуміло, що виходячи з найпростіших функцій та застосовуючи оператори суперпозиції, примітивної рекурсії та мінімізації, процес отримання нових функцій можна продовжувати необмежено. Функції, які отримаємо при цьому, називають рекурсивними.

Означення 6.2.1. *Часткова функція f називається частково рекурсивною, якщо вона може бути одержана з найпростіших функцій за допомогою скінченного числа вживань операторів суперпозиції, примітивної рекурсії та мінімізації. Інакше ка-*

жучи, f — рекурсивна, якщо існує така скінченна послідовність $g_1, \ldots, g_n$, часткових функцій, що $g_n = f$ та кожна функція цієї послідовності або є найпростішою, або одержана з попередніх за допомогою операторів суперпозиції, примітивної рекурсії та мінімізації.

Клас частково рекурсивних функцій ширший класу примітивно рекурсивних функцій, оскільки всі примітивно рекурсивні функції є всюди визначеними, а серед частково рекурсивних функцій зустрічаються функції не всюди визначені, а також ніде не визначені.

Поняття частково рекурсивних функцій є одним з головних понять теорії алгоритмів. Його значення полягає в тому що з одного боку, кожна стандартно задана частково рекурсивна функція обчислювана за деякою процедурою механічного характеру, що відповідає інтуїтивному представленню про алгоритми. З іншого боку, які б класи точно описаних алгоритмів не будувались, у всіх випадках з'ясовувалось, що обчислювані за ними числові функції є частково рекурсивними.

Якщо функція усюди визначена і рекурсивна, то вона називається *загально рекурсивною*.

Отже, *кожна загально рекурсивна функція рекурсивна*. Крім того, функція, яку отримано за допомогою оператора примітивної рекурсії, визначена для всіх значень аргументів і тому є загально рекурсивною (якщо до того ж її побудова почата з найпростіших функцій).

Всі функції, розглянуті в прикладах 6.2.1−6.2.7, рекурсивні, а 6.2.3−6.2.5 ще й загально рекурсивні.

При побудові аксіоматичної теорії висловлень початкові формули (аксіоми) та правила виведення вибиралися так, щоб одержані в теорії формули вичерпали б усі тавтології алгебри висловлень. Чого ж прагнемо в теорії рекурсивних функцій, чому саме так вибрали найпростіші функції та оператори для отримання нових функцій? Рекурсивними функціями прагнемо вичерпати всі можливі функції, що піддаються обчисленню за допомогою якої-небудь певної процедури механічного характеру. В теорії рекурсивних функцій природно висувається відповідна наукова гіпотеза, що носить назву тези Чорча.

Теза Чорча. *Числова функція тоді і тільки тоді алгоритмічно (або машинно) обчислювана, коли вона рекурсивна.*

Ця гіпотеза також не може бути доведена строго математично, вона підтверджується практикою, досвідом, бо покликана пов'язати практику і теорію. Всі конкретні функції, що розглядалися в математиці, які визнаються обчислюваними в інтуїтивному значенні, виявлялися рекурсивними.

Вправи

1. Довести, що функція $f(x, y) = x + y$ є примітивно рекурсивною.
2. Довести, що функція $f(x, y) = xy$ є примітивно рекурсивною.
3. Довести, що функція $f(x, y) = x \div y$ є примітивно рекурсивною.
4. Довести, що функція $f(x, y) = min\{x, y\}$ є примітивно рекурсивною.
5. Довести, що функція $f(x, y) = max\{x, y\}$ є примітивно рекурсивною.
6. Довести, що логічна функція заперечення $f(x) = \neg x$ є примітивно рекурсивною.
7. Довести, що логічна функція кон'юнкція $f(x, y) = x \wedge y$ є примітивно рекурсивною.
8. Довести, що логічна функція диз'юнкція $f(x, y) = x \vee y$ є примітивно рекурсивною.
9. Довести, що предикат $P_1(x, y)$ −"x ділиться націло на y" є примітивно рекурсивним.
10. Довести, що предикат $P_2(x, y)$ −"$x < y$" є примітивно рекурсивним.

Розділ 7

АЛГОРИТМ ЯК АБСТРАКТНА МАШИНА

7.1. ЗАГАЛЬНІ ПОНЯТТЯ

Одним з можливих розв'язків задачи уточнення поняття алгоритму є точний опис класу частково рекурсивних функцій разом з тезою Чорча. Проте цей розв'язок не зовсім прямий, оскільки поняття обчислюваної функції є вторинним по відношенню до поняття алгоритму. Виникає питання, а чи можна уточнити безпосередньо саме поняття алгоритму та вже потім за його допомогою визначити точно також й клас обчислюваних функцій? Такий напрямок думки привів до побудови іншого, ніж рекурсивні функції, класу моделей алгоритму. Основна ідея полягає в тому, що алгоритмічні процеси це процеси, які може здійснювати певним чином налаштована машина, що моделює виконання окремих операцій людиною. Робота такої машини і є виконання певного алгоритму. Зважаючи на властивості алгоритму (п. 5.1), можна викласти загальні вимоги до таких машин:

1) характер їх функціювання повинен бути *дискретним*, тобто складатись з окремих кроків, кожен з яких виконується тільки по завершенню попереднього;
2) дії повинні бути *детермінованими*, тобто кроки виконуються у строгому порядку, а їх результат визначається самим кроком та результатами попередніх кроків;
3) на початку роботи машині надаються *вихідні дані* з області означення алгоритму;
4) результат повинен бути отриманим за *скінчену* кількість кроків (або інформацію, що вважати результатом);
5) машина повинна бути *універсальною*, тобто такою, що б з її допомогою можна було би виконати будь-який алгоритм.

Зрозуміло, що чим простіше структура машини та чим елементарніші її кроки, то тим більше підстав вважати що її робота і є виконання алгоритму. Для відповіді на запитання, які кроки роботи машини вважати елементарними, потрібно вернутись до питання перетворення інформації, що представлена за допомогою деякого скінченого алфавіту. Вимога скінченності алфавіту є законним наслідком тих обставин, що розв'язок мусить бути отриманим за скінчену кількість кроків. Якщо інформація не представлена в дискретній формі, то її обробка загалом може містити нескінченну кількість кроків.

Нехай вихідні дані з області визначення алгоритму представлені за допомогою алфавіту A та утворюють скінченну послідовність знаків $\{a_1, a_2, \dots a_n\}$, така послідовність називається *словом*. У результаті виконання алгоритму утворюється нове слово $\{b_1, b_2, \dots b_m\}$, подане, взагалі кажучи, в іншому алфавіті B. З першого погляду для проведення такого перетворення у якості елементарних визначаються такі операції (кроки):

1) заміна одного знаку вихідного слова a_i знаком b_j з алфавіту B;
2) видалення знаку вихідного слова;
3) додавання до вихідного слова знаку b_j з алфавіту B.

Отже, алгоритм виявляється *скінченою послідовністю дій,* що виконуються над даними, які представлені за допомогою скінченого алфавіту. Тобто має місце таке означення алгоритму: *алгоритм це скінчена система правил перетворення інформації (даних) над довільним скінченим алфавітом.*

Слід зауважити, якщо до алфавіту включено знак, що має значення порожнього знаку, додання якого до слова зліва або справа не змінює цього слова, то зрозуміло, що операція (2) є заміною a_i порожнім знаком. А операція (3) є заміною порожнього знаку знаком b_j. Отже, всі можливі алфавітні перетворення зводяться до операції (1), тобто заміні одного знаку вихідного слова іншим знаком. Саме тому функціонування абстрактної машини зводиться до того, що вона оглядає символи (зчитує та розпізнає їх), що записані в пам'яті (якою є нескінчена стрічка), та в залежності від стану в якому вона перебуває і того, який символ оглядає, вона замінює його іншим символом, після чого вона переходить в новий стан, читає наступний символ і так далі до команди про зупинення роботи. Оскільки такі машини є чисто модельною, теоретичною побудовою, вони отримали назву абстрактних машин та розглядаються як одна з можливих алгоритмічних систем.

За дев'ять років до появи першої ЕОМ з'явилося поняття машини Т'юринга, названої так за ім'ям англійського математика, що сформулював його в 1937 році. Це представлення було однією з перших і вельми вдалих спроб дати точний математичний еквівалент для загального інтуїтивного поняття алгоритму.

7.2. ПОНЯТТЯ ПРО МАШИНИ Т'ЮРИНГА

Машина Т'юринга є математичною машиною (абстрактною), а не машиною фізичною. Вона є таким же математичним об'єктом, як визначник, матриця, функція, похідна, інтеграл і т.д. Також, як і інші математичні поняття, поняття машини Т'юринга відображає об'єктивну реальність, моделює реальні процеси. Саме Т'юринг зробив спробу змоделювати дії людини (математика), що здійснює якусь розумову творчу діяльність. Така людина, перебуваючи в певному «умонастрої», проглядає деякий текст. Потім вона вносить в цей текст якісь зміни, змінює «умонастрій» і переходить до перегляду подальших записів.

Приблизно так само діє машина Т'юринга. Її зручно уявляти у вигляді автоматично працюючого пристрою. У кожен дискретний момент часу пристрій, знаходячись в деякому стані, оглядає вміст однієї комірки на стрічці, що протягується через пристрій, та робить крок, який полягає в тому, що пристрій переходить в новий стан, змінює вміст комірки і переходить до огляду наступної комірки, справа або зліва. Причому крок здійснюється на підставі записаної команди. Сукупність всіх команд є програмою машини Т'юринга.

Тепер дамо детальний опис *машини Т'юринга* $\mathfrak{T}$. В своєму розпорядженні машина $\mathfrak{T}$ має скінченне число знаків (символів, літер), що створюють так званий зовнішній алфавіт $A = \{a_0, a_1, ..., a_n)$. У кожну комірку стрічки, в кожен дискретний момент часу може бути записаний тільки один символ з алфавіту A. Задля одноманітності зручно вважати, що серед літер зовнішнього алфавіту A є «порожня літера» і саме вона записана в порожній комірці стрічки. Домовимося, що «порожньою літерою» або символом порожньої комірки є літера a_0. Стрічка передбачається необмеженою в обидві сторони, але в кожен момент часу на ній записане скінченне число непорожніх літер.

Машина $\mathfrak{T}$ в кожен момент часу може знаходитися в одному із скінченного числа внутрішніх станів, сукупність яких $Q = \{q_0, q_1, ..., q_m\}$. Серед станів виділяють два – початковий q_1 та завершальний, або стан зупинки, q_o. Знаходячись в стані q_1, машина починає працювати. Потрапивши в стан q_o, машина зупиняється.

Машина $\mathfrak{T}$ працює за *програмою* (*функціональною схемою*). Програма складається з команд. Кожна команда $T(i, j)$ (i = 1, 2, ..., m; j = 0, 1, ..., n) є виразом одного з таких видів:

$$q_ia_j \to q_ka_lC,\ q_ia_j \to q_ka_lП,\ q_ia_j \to q_ka_lЛ, \qquad (7.2.1)$$

де $0 \le k \le m,\ 0 \le l \le n$. У виразах першого типу символ C часто опускатимемо.

Як саме працює машина Т'юринга? Знаходячись в який-небудь момент часу не в стані зупинки (тобто в стані відмінному від q_0), машина здійснює крок, який повністю визначається її поточним станом q_i, та символом a_j, що сприймається нею в даний момент на стрічці. При цьому зміст кроку регламентовано відповідною командою $T(i, j): q_i a_j \to q_k a_l X$, *де* $X \in \{C, П, Л\}$. Крок полягає в тому, що:

1) вміст a_j комірки на стрічці стирається, та на його місце записується символ a_l (який може співпадати з a_j);
2) машина переходить в новий стан q_k (він також може співпадати з попереднім станом q_i);
3) машина переходить до огляду наступної правої комірки від тієї, яку оглядала щойно, якщо $X = П$, або до огляду наступної лівої комірки, якщо $X = Л$, або ж продовжує оглядати ту ж комірку стрічки, якщо $X = C$.

У наступний момент часу (якщо $q_k \neq q_o$) машина робить крок, регламентований командою $T(k, l)$: $q_k\ a_l \to q_r\ q_s X$, і т.д.

Оскільки, за умовою, робота машини $\mathfrak{T}$ повністю визначається її станом q_i в даний момент та вмістом a_j комірки, що розглядається у цей момент, то для кожних q_i, a_j ($i = 1, 2, ..., m; j = 0, 1, ..., n$) програма машини повинна містити одну і лише одну команду, що починається символами $q_i a_j$. Тому програма машини Т'юринга із зовнішнім алфавітом $A = \{a_0, a_1 ..., a_n\}$ та алфавітом внутрішніх станів $Q = \{q_0, q_1, ..., q_m\}$ містить $m(n + 1)$ команд.

Як відомо, *словом* в деякому алфавіті A називається будь-яка скінченна послідовність літер цього алфавіту. Під k-ою *конфігурацією* розумітимемо зображення стрічки машини з інформацією, що склалася на ній до початку k-го кроку (або слово в алфавіті A, записане на стрічку до початку k-го кроку), з вказівкою того, яка комірка оглядається на цьому кроці та в якому стані знаходиться машина. Мають сенс лише скінченні конфігурації, тобто такі, в яких всі коміркі стрічки, за виключенням, можливо, скінченного числа, порожні. Конфігурація називається *завершальною*, якщо стан, в якому при цьому знаходиться машина — стан зупинки.

Якщо обрати за початкову яку-небудь конфігурацію машини $\mathfrak{T}$, що не є завершальною, то робота машини полягатиме в тому, щоб послідовно, крок за кроком, перетворювати початкову конфігурацію відповідно до програми машини до тих пір, поки не буде досягнута завершальна конфігурація. Після цього робота машини Т'юринга вважається такою, що закінчилася, а результатом роботи вважається досягнута завершальна конфігурація.

Вважаємо, що непорожнє слово α в алфавіті $A \backslash \{a_0\} = \{a_1, ..., a_n\}$ сприймається машиною в стандартному положенні, якщо воно записане в послідовних комірках стрічки, всі інші комірки порожні, та машина оглядає крайню справа комірку з тих, в яких записане слово α. Стандартний стан називається *початковим* (*завершальним*), якщо машина, що сприймає слово в стандартному положенні, знаходиться в початковому стані q_1 (відповідно, в стані зупинки q_0). Нарешті, вважаємо, що *слово* α *переробляється машиною в слово* β, якщо від слова α, що сприймається в початковому стандартному положенні, машина після виконання скінченного числа команд приходить до слова β, що сприймається в положенні зупинки.

Введені поняття, пов'язані з машинами Т'юринга, проілюструємо на прикладах.

Приклад 7.2.1. Потрібно побудувати машину Т'юринга, яка додає одиницю до числа на стрічці. Вхідне слово складається з цифр цілого десяткового числа, записаних в послідовні комірки на стрічці. У початковий момент машина знаходиться проти самої правої цифри числа.

Розв'язок. Машина має додати одиницю до останньої цифри числа. Якщо остання цифра дорівнює 9, то її замінити на 0 та додати одиницю до попередньої цифри. Програма для даної машини Т'юрінга може виглядати так:

	a_0	0	1	2	3	4	5	6	7	8	9
q_1	$1Сq_0$	$1Сq_0$	$2Сq_0$	$3Сq_0$	$4Сq_0$	$5Сq_0$	$6Сq_0$	$7Сq_0$	$8Сq_0$	$9Сq_0$	$0Лq_1$

У цій машині Т'юринга зовнішній алфавіт A = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, q_1 — стан зміни цифри, q_0 — стан зупинки. Якщо в стані q_1 автомат бачить цифру 0.. 8, то він замінює її на 1.. 9 відповідно та переходить в стан q_0, тобто машина зупиняється. Якщо ж він бачить цифру 9, то замінює її на 0, зсувається вліво, залишаючись у стані q_1. Так продовжується до тих пір, поки автомат не зустріне цифру менше 9. Якщо ж всі цифри були рівні 9, то він замінить їх нулями, запише 0 на місці старшої цифри, зрушиться вліво та в порожній клітині запише 1. Потім перейде в стан q_0, тобто зупиниться.

Приклад 7.2.2. Нехай маємо машину Т'юринга із зовнішнім алфавітом A ={ 0, 1}, алфавітом внутрішніх станів $Q = \{q_0, q_1, q_2\}$ і з такою функціональною схемою (програмою):

$$q_1 0 \to q_2 0П,\ q_2 0 \to q_0 1,\ q_1 1 \to q_1 1П,\ q_2 1 \to q_2 1П.$$

Знайдемо, в яке слово переробить ця машина слово 101, виходячи із стандартного початкового стану. Послідовно виписуватимемо конфігурації машини при переробці нею цього слова. Отримаємо стандартний початковий стан:

					q_1				
(1)			1	0	1				

На першому кроці діє команда: $q_1 1 \to q_1 1П$. У результаті на машині утворюється така конфігурація:

						q_1			
(2)			1	0	1	0			

На другому кроці діє команда: $q_1 0 \to q_2 0П$ і на машині утворюється конфігурація:

					q_1		q_2		
(3)			1	0	1	0	0		

Нарешті, третій крок обумовлений командою: $q_2 0 \to q_0 1$. У результаті цього утворюється конфігурація:

							q_0		
(4)			1	0	1	0	1		

Ця конфігурація є завершальною, тому що машина опинилася в стані зупинки q_0.

Отже, початкове слово 101 перероблене машиною в слово 10101.

Одержану послідовність конфігурацій можна записати коротше. Конфігурація (1) записується у вигляді такого слова в алфавіті $A \cup Q$: $10q_1 1$ (вміст комірки, що розглядається, записаний праворуч від стану, в якому знаходиться в даний момент машина). Далі, конфігурація (2) записується так: $101q_1 0$, конфігурація (3) — $1010q_2 0$, і нарешті, (4) — $1010q_0 1$. Вся послідовність записується так:

$$10q_1 1 \Rightarrow 101q_1 0 \Rightarrow 1010q_2 0 \Rightarrow 1010q_0 1.$$

Наведемо послідовність конфігурацій при переробці цією машиною слова 11011, виходячи з початкового стану, при якому в стані q_1 оглядається крайня ліва комірка, в якій міститься символ цього слова:

			q_1						
(1)			1	1	0	1	1		

				q_1					
(2)			1	1	0	1	1		

					q_1				
(3)			1	1	0	1	1		

						q_2			
(4)			1	1	0	1	1		

							q_2		
(5)			1	1	0	1	1		

								q_2	
(6)			1	1	0	1	1	0	

								q_0	
(7)			1	1	0	1	1	1	

Коротший запис цієї послідовності конфігурацій, тобто процесу роботи машини буде

$$q_1 11011 \Rightarrow 1q_1 011 \Rightarrow 11q_1 011 \Rightarrow 110q_2 11 \Rightarrow$$
$$\Rightarrow 1101q_2 1 \Rightarrow 11011q_2 0 \Rightarrow 11011q_0 1.$$

Отже, слово 11011 перероблене машиною в слово 110111.

Зробимо деякі висновки. Такий детально проведений опис будови цієї машини (розбита на комірки стрічка, зчитуюча головка) зрештою не має сенсу. Машина Т'юринга це деяке правило (алгоритм) для перетворення слів алфавіту $A \cup Q$, тобто конфігурацій. Отже, для означення машини Т'юринга потрібно задати її зовнішній та внутрішній алфавіти, програму та вказати, які з символів позначають порожню комірку і заключний стан.

Створення (синтез) машин Т'юринга (тобто написання відповідних програм) є задачею значно складнішою, ніж процес застосування даної машини до даних слів.

7.3. ФУНКЦІЇ, ОБЧИСЛЮВАНІ ЗА Т'ЮРИНГОМ

Попередньо ми вже говорили про обчислювані функції в припущенні, що поняття алгоритму нам зрозуміле інтуїтивно. Тепер ми озброєні точним означенням алгоритму. Алгоритм для нас тепер — машина Т'юринга; алгоритмічний процес — процес роботи машини Т'юринга. Сформулюємо таке означення.

ОЗНАЧЕННЯ 7.3.1. *Функція називається обчислюваною за Т'юрингом, якщо існує машина Т'юринга, що обчислює її.*

Залишається домовитися про деякі умовності для того, щоб це означення стало повністю точним.

Нагадаємо, спершу, що йдеться про функції (або можливо про часткові функції, тобто не усюди визначені), задані на множині натуральних чисел та такі, що набувають також натуральні значення. Відтак, потрібно домовитись, як записувати на стрічці машини Т'юринга значення $x_1, x_2, ..., x_n$ аргументів функції $f(x_1, x_2, ..., x_n)$, з якого стану починати переробку початкового слова і, нарешті, в якому положенні діставати значення функції. Це можна робити, наприклад, так. Значення $x_1, x_2, ..., x_n$ аргументів розташовуватимемо на стрічці у вигляді слова:

$$\underbrace{01\ldots1}_{x_1}\underbrace{01\ldots1}_{x_2}0\ldots\underbrace{01\ldots1}_{x_n}0.$$

Починати переробку цього слова будемо зі *стандартного початкового положення*, тобто зі стану, при якому в стані q_1

розглядається крайня права одиниця записаного слова. Якщо функція $f(x_1, x_2, ..., x_n)$ визначена на даному наборі значень аргументів, то в результаті на стрічці повинно бути записано підряд $f(x_1, x_2, ..., x_n)$ одиниць; інакше машина повинна працювати нескінченно. При виконанні всіх перерахованих умов говоритимемо, що *машина Т'юринга обчислює дану функцію f.* Отже, сформульоване означення 7.3.1 стає абсолютно точним.

Сконструювати машину Тьюринга — означає написати (скласти) її програму. У цьому процесі два етапи: спочатку *створити алгоритм* обчислювання значень функції, а потім *записати його на мові машини Тьюринга (запрограмувати)*.

7.4. ТЕЗА Т'ЮРИНГА

Повернемося до інтуїтивного поняття алгоритму (див. пункт 5.1). Нагадаємо, що одна з властивостей алгоритму полягає в тому, що він є єдиним способом, який дозволяє для кожної задачі з деякої нескінченної множині задач за скінченне число кроків знайти її розв'язок.

На поняття алгоритму можна поглянути і з дещо іншої точки зору. Кожну задачу з нескінченної множини задач можна подати (закодувати) деяким словом того ж алфавіту. У результаті одержимо функцію, що задана на деякій підмножині множини всіх слів вибраного алфавіту та яка набуває значень в множині всіх слів того ж алфавіту. Розв'язати деяку задачу — означає знайти значення цієї функції на слові, що кодує дану задачу. А мати алгоритм для вирішення всіх задач даного класу — означає мати єдиний спосіб, що дозволяє за скінченне число кроків «обчислювати» значення побудованої функції для будь-яких значень аргументу з її області визначення. Отож, *алгоритмічна проблема* — по суті проблема про обчислення значень функції, заданої в деякому алфавіті.

Лишається з'ясувати, що означає вміти обчислювати значення функції. Це означає обчислювати значення функції за допомогою відповідної машини Т'юринга. Для яких же функцій можливе їх обчислення за Т'юрингом? Численні розвідки вчених, чималий досвід показали, що такий клас функцій надзвичайно широкий. Кожна функція, для обчислення значень якій існує який-небудь алгоритм, виявлялася обчислюваною за допомогою деякої машини Т'юринга. Це дало Т'юрингу привід висловити гіпотезу, яку називають основною гіпотезою теорії алгоритмів або тезою Т'юринга.

Теза Т'юринга*. Для знаходження значень функції, заданої в деякому алфавіті, тоді і тільки тоді існує який-небудь алгоритм, коли функція є обчислюваною за Т'юрингом, тобто коли її можна обчислювати на відповідній машині Т'юринга.*

Звідси випливає, що строго математичне поняття обчислюваної (за Т'юрингом) функції є, по суті, ідеальною моделлю взятого з досвіду поняття алгоритму. Ця теза є ні що інше, як аксіома, постулат, про взаємозв'язки досвіду з тією математичною теорією, яку під цей досвід хочемо підвести. Звичайно ж, ця теза у принципі не може бути доведена методами математики, тому що вона не має внутрішньо математичного характеру (одна сторона у тезі, а саме, поняття алгоритму — не є точним математичним поняттям). Вона висловлена, виходячи з досвіду, і саме досвід підтверджує її дієздатність. Так само, наприклад, не можуть бути доведені і математичні закони механіки: вони відкриті Ньютоном та багатократно підтверджені досвідом.

Втім, не виключена принципова можливість спростування тези Т'юринга. Для цього повинна бути вказана функція, обчислювана за допомогою якого-небудь алгоритму, але не обчислювана на жодній машині Т'юринга. Але така нагода вбачається маловірогідною, — у цьому одне із значень гіпотези, — всякий алгоритм, який буде відкритий, може бути реалізований на машині Т'юринга.

7.5. МАШИНА Т'ЮРИНГА ТА ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНІ МАШИНИ

У машині Т'юринга розчленування (аналіз) обчислювального процесу на найпростіші операції доведено до граничної можливості: розпізнавання єдиного розглянутого входження символу, переміщення точки нагляду даного ряду символів в сусідню точку та зміну інформації, що є в пам'яті. Звичайно, таке потужне дроблення обчислювального процесу, значно його здовжує. Проте, логічна структура процесу, розчленованого, так би мовити, до атомарного стану, спрощується і постає в деякому стандартному вигляді, вельми зручному для теоретичних досліджень. Саме таке розчленовування на найпростіші складові обчислювального процесу на машині Т'юринга дає ще один непрямий аргумент на користь тези Т'юринга: всяка функція, обчислювана за допомогою деякого алгоритму, може бути обчислена на машині Т'юринга, тому що кожний крок даного алгоритму можна розчленувати на ще більш дрібні операції, які реалізуються в машині Т'юринга. Зрозуміло, вивчення машин Т'юринга і практика складання програм для них закладають фундамент алгоритмічного мислення, сенс якого полягає в тому, що потрібно вміти розділити той або інший процес обчислення на прості складові кроки. Тому поняття машини Т'юринга є теоретичним інструментом аналізу алгоритмічного процесу, а отже, аналізу суті алгоритмічного мислення.

Алгоритмічний процес у сучасних комп'ютерах розчленовано на не такі дрібні складові, як в машинах Т'юринга. Так, для виконання операції додавання на машині Т'юринга складається ціла програма, а в сучасному комп’ютері така операція є елементарною.

Машина Т'юринга також має нескінченну зовнішню пам'ять (необмежена в обидві сторони стрічка, розбита на комірки). Але в жодній реально існуючій машині нескінченної пам'яті бути не може. Це говорить про те, що машини Т'юринга відображають потенційну можливість необмеженого збільшення об'єму пам'яті сучасних ЕОМ.

Можна провести більш докладний порівняльний аналіз роботи сучасного комп’ютера та машини Т'юринга. У більшості ЕОМ прийнята триадресна система команд, обумовлена необхідністю виконання бінарних операцій, в яких бере участь вміст трьох елементів пам'яті. Наприклад, число з комірки a помножається на число з комірки b, та результат відправляється в комірку c. Існують комп’ютери двоадресні та одноадресні. Зокрема, одноадресна система ЕОМ працює так: викликається (в суматор) число з комірки a ; в суматорі відбувається, наприклад, множення цього числа на число з комірки b; результат відправляється на суматор в комірку c. Машину Т'юринга можна вважати одноадресною машиною, в якій система одноадресних команд спрощена ще більше: на кожному кроці роботи машини команда наказує заміну лише єдиного знака, що зберігається в комірці, яка оглядається, а адреса комірки, яка оглядається, при переході до наступного такту може мінятися лише на одиницю (огляд сусідньої зліва або справа комірки стрічки) або не міняється зовсім. Це подовжує процес, проте різко уніфікує його, робить стандартним.

Можемо сказати, підводячи підсумки, що сучасні комп’ютери є деякі реальні фізичні моделі машини Т'юринга, огрублені з погляду теорії, але створені з метою реалізації конкретних обчислювальних процесів. У свою чергу, поняття машини Т'юринга та теорія таких машин є теоретичним фундаментом і обґрунтовуванням сучасних ЕОМ.

Вправи

1. Скласти програму машини Т'юринга додавання двох чисел $x + y$.
2. Скласти програму машини Т'юринга для копіювання числа x.
3. Скласти програму машини Т'юринга для копіювання слова без крайніх символів.
4. Скласти програму машини Т'юринга для підрахунку букв у прізвищі.
5. Скласти програму машини Т'юринга що переставляє ім'я та прізвище.
6. Скласти програму машини Т'юринга що додає цифру 2 у десятковий запис після останньої цифри числа x.
7. Відняти два натуральні числа, що подані в десятковій системі числення.
8. Знайти найбільший спільний дільник двох натуральних чисел, що подані в десятковій системі числення.

Розділ 8

ІНШІ УТОЧНЕННЯ ПОНЯТТЯ АЛГОРИТМУ

8.1. НОРМАЛЬНІ АЛГОРИТМИ МАРКОВА

Як відомо, поняття машини Т'юринга — не єдиний відомий шлях уточнення поняття алгоритму. Тут стисло розглянуто нормальні алгоритми Маркова — ще один із способів такого уточнення.

У кінці 40-х — початку 50-х років XX-го століття математиком А. А. Марковим (1903-1979) була розроблена теорія нормальних алгоритмів (або алгорифмів, як називав їх творець теорії). Ці алгоритми є деякими правилами переробки слів в якомусь алфавіті, так що початкові дані та шукані результати для алгоритмів є словами в деякому алфавіті.

Як і у випадку машин Т'юринга, *алфавітом* називають будь-яку непорожню множину. Його елементи називаються *літерами,* а будь-які скінченні послідовності літер — *словами* в даному алфавіті. Для зручності міркувань допускаються порожні слова (вони не мають в своєму складі жодної літери). Порожнє слово позначають Λ. Якщо A та B — два алфавіти, причому $A \subseteq B$, то B називають *розширенням* алфавіту A.

Латинськими літерами P, Q, R, ... (або цими ж літерами з індексами) позначають слова. Одне слово може бути складовою частиною іншого слова. Тоді перше називається *підсловом* другого або входженням в друге. Наприклад, якщо A — алфавіт кирилиця, то можемо розглянути такі слова: $P_1 =$ *параграф*, $P_2 =$ *граф*, $P_3 =$ *ра*. Слово P_2 є підсловом слова P_1, а P_3 — підсловом слова P_1 та P_2, причому, в P_1 воно входить двічі.

ОЗНАЧЕННЯ 8.1.1. *Підстановкою Маркова називається операція над словами, що задається за допомогою впорядкованої пари слів (P, Q). Ця підстановка зводиться до такого. В заданому слові R, знаходять перше входження слова P (якщо таке є) та, не змінюючи решти частин слова R, замінюють в ньому це входження словом Q. Одержане слово називається результатом застосування підстановки Маркова (P, Q) до слова R. Якщо ж першого входження P в слово R нема, отже, взагалі немає жодного входження P в R, то приймається, що підстановка Маркова (P,Q) не застосовна до слова R.*

Окремими випадками підстановок Маркова є підстановки з порожніми словами: (Λ, Q), (P, Λ) (Λ, Λ).

Приклад 8.1.1. В таблиці розглядаються приклади підстановок Маркова. У кожному її рядку спочатку дається слово, що перетворюється, потім вживана до нього підстановка Маркова і нарешті — слово, що виходить в результаті.

Слово, що перетворюється	Підстановка Маркова	Результат
169163441716	(163441, 11)	16911716
карбованець	(карбов, бр)	бранець
логіка	(іка, Λ)	лог
пара	(ар, ор)	пора
віче	(так, ні)	[не застосовна]

Для позначення підстановки Маркова (P, Q) використовується запис $P \to Q$. Вона називається *формулою підстановки* (P, Q). Деякі підстановки (P, Q) називатимемо *заключними* (значення назви стане зрозуміле згодом). Для позначення таких підстановок використовуватимемо запис $P \to \cdot Q$, називаючи її формулою завершальної підстановки. Слово P називається *лівою частиною*, а Q — *правою частиною* у формулі підстановки.

Впорядкований скінченний список формул підстановок в алфавіті A:

$$\begin{cases} P_1 \to (\cdot)Q_1 \\ P_2 \to (\cdot)Q_2 \\ \dots\dots\dots\dots \\ P_k \to (\cdot)Q_k \end{cases}$$

— називається схемою (або записом) нормального алгоритму в A. (Запис крапки в дужках означає, що вона може стояти в цьому місці, а може бути відсутня.) Дана схема визначає (детермінує) алгоритм перетворення слів, званий нормальним алгоритмом Маркова. Наведемо його точне означення.

Нормальним алгоритмом (Маркова) в алфавіті A називається таке правило побудови послідовності P_i, слів в алфавіті A, виходячи з даного слова P в цьому алфавіті. Як початкове слово P_0 послідовності береться слово P. Нехай для деякого $i > 0$ слово P_i побудовано та процес побудови даної послідовності ще не скінчився. Якщо при цьому в схемі нормального алгоритму немає формул, ліві частини яких входили б в P_i, то P_{i+1} визнають рівним P_i а процес побудови послідовності вважається завершеним. Якщо ж в схемі є формули з лівими частинами, що входять в P_i, то за P_{i+1} береться результат підстановки Маркова правої частини першої з таких формул замість першого входження її лівої частини в слово P_i. Процес побудови послідовності вважається *завершеним*, якщо на даному кроці була застосована формула заключної підстановки, та таким, що продовжується в іншому випадку. Якщо процес побудови згаданої послідовності обривається, то говорять, що даний *нормальний алгоритм застосовний до слова P*. Останній член Q послідовності називається *результатом застосування нормального алгоритму до слова P*. Говорять, що *нормальний алгоритм переробляє P в Q*.

Послідовність P_i записуватимемо так:

$$P_0 \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \ldots \Rightarrow P_{m-1} \Rightarrow P_m,$$

де $P_0 = P$ та $P_m = Q$.

Наведемо приклади нормальних алгоритмів.

Приклад 8.1.2. Нехай $A = \{a, b\}$ — алфавіт. Розглянемо таку схему нормального алгоритму в A:

$$\begin{cases} a \to \cdot\Lambda, \\ b \to b. \end{cases}$$

Легко зрозуміти, як працює нормальний алгоритм, що визначений цією схемою. Всяке слово P в алфавіті A, що містить хоча б одне входження літери a, він переробляє в слово, що виходить з P викреслюванням в ньому найлівішого (першого) входження літери a. Порожнє слово він переробляє в порожнє. Алгоритм незастосовний до таких слів, які містять тільки літеру b. Наприклад, $aabab \Rightarrow abab$, $ab \Rightarrow b$, $aa \Rightarrow a$, $bbab \Rightarrow bbb$, $baba \Rightarrow bba$.

Приклад 8.1.3. Нехай $A = \{a_0, a_1, \ldots, a_n\}$ — алфавіт. Схема така

$$\begin{cases} a_0 \to \Lambda \\ a_1 \to \Lambda \\ \ldots\ldots\ldots\ldots \\ a_n \to \Lambda \\ \Lambda \to \cdot\Lambda. \end{cases}$$

Вона визначає нормальний алгоритм, що переробляє всяке слово (в алфавіті A) в порожнє слово. Наприклад

$$a_1a_2a_3a_2a_0 \Rightarrow a_1a_2a_3a_2 \Rightarrow a_2a_3\ a_2 \Rightarrow a_3a_2 \Rightarrow a_3 \Rightarrow \Lambda \Rightarrow \Lambda;$$

$$a_3a_2a_0a_1a_2a_3a_2 \Rightarrow a_3a_2a_1a_2a_3\ a_2 \Rightarrow a_3a_2a_2a_3a_2 \Rightarrow a_3a_2a_3a_2 \Rightarrow a_3a_3a_2 \Rightarrow$$
$$\Rightarrow a_3a_3 \Rightarrow a_3 \Rightarrow \Lambda \Rightarrow \Lambda.$$

Наведемо приклад нормального алгоритму що обчислює функцію.

Приклад 8.1.4. Дано функцію:

$$\varphi_3(11\ldots1) = \begin{cases} 1, & \text{якщо } n \text{ ділиться на 3;} \\ 0, & \text{якщо } n \text{ не ділиться на 3.} \end{cases}$$

Розглянемо нормальний алгоритм в алфавіті $A = \{1\}$ з такою схемою:

$$\begin{cases} 111 \to \Lambda \\ 11 \to \cdot\Lambda \\ 1 \to \cdot\Lambda \\ \Lambda \to \cdot 1. \end{cases}$$

Цей алгоритм працює так: поки число літер 1 в слові не менше 3, алгоритм послідовно стирає по три літери. Якщо число літер менше 3, але більше 0, то літери 1 або 11, що залишилися, стираються заключно; якщо слово порожнє, воно заключно переводиться в слово 1.

Наприклад:

$$1111111 \Rightarrow 1111 \Rightarrow 1 \Rightarrow \Lambda;\ 111111111 \Rightarrow$$
$$\Rightarrow 111111 \Rightarrow 111 \Rightarrow \Lambda \Rightarrow 1.$$

Отже, розглянутий алгоритм реалізує (або обчислює) дану функцію. Сформулюємо тепер точне означення обчислюваності функцій.

ОЗНАЧЕННЯ 8.1.2*. Функція f, задана на деякій множині слів алфавіту A, називається нормально обчислюваною, якщо знайдеться таке розширення B даного алфавіту ($A \subseteq B$) і такий нормальний алгоритм у B, що кожне слово P (в алфавіті A) з області визначення функції f цей алгоритм переробляє в слово f(P).*

Отже, нормальний алгоритм прикладу 8.4 показує, що функція φ_3 нормально обчислювана. Причому, відповідний нормальний алгоритм вдалося побудувати в тому ж самому алфавіті A, на словах якого були задані функції, що розглядалися, тобто розширювати алфавіт не було потреби ($B = A$).

Творець теорії нормальних алгоритмів математик А.А. Марков висунув гіпотезу, що одержала назву *принцип нормалізації Маркова*.

Принцип нормалізації Маркова*. Для знаходження значень функції, заданої в деякому алфавіті, тоді і тільки тоді існує який-небудь алгоритм, коли функція нормально обчислювана.*

Цей принцип носить позаматематичний характер і не може бути строго доведений. Його сформульовано на підставі математичного та практичного досвіду людства. Все, що в попередньому параграфі було сказане про тезу Т'юринга, можна з повним правом віднести до принципу нормалізації Маркова.

8.2. ЕКВІВАЛЕНТНІСТЬ РІЗНИХ ПІДХОДІВ ДО ПОНЯТТЯ АЛГОРИТМУ

Ми ознайомилися з декількома теоріями, кожна з яких уточнює поняття алгоритму. Виникає питання, як пов'язані ці теорії між собою. Відповідь на це дає така теорема.

Теорема 8.2.1. *Співпадають такі класи функцій (що задані на множині натуральних чисел і приймають натуральні значення):*

а) клас усіх функцій, обчислюваних за Т'юрингом;
б) клас усіх нормально обчислюваних функцій;
в) клас усіх рекурсивних функцій.

Це вже не гіпотеза і не теза, а математична теорема, яка цілком строго може бути доведена. Проте опустимо її доведення, вважаючи, що доведення теорем, які приводяться, менш важливі, ніж їх формулювання, а формулювання теорем менш важливі, ніж означувані поняття.

Дуже корисно з'ясувати сенс та значення наведеної теореми. Вона означає, що теорії машин Т'юринга, нормальних алгоритмів і рекурсивних функцій рівносильні. В різний час у різних країнах вчені незалежно один від одного, вивчаючи інтуїтивне поняття алгоритму та алгоритмічної обчислюваності, створили теорії, що описують дане поняття, які виявилися рівносильними. Якби один з цих класів виявився ширший за який-небудь інший клас, то відповідна теза Т'юринга, Маркова або Чорча була б спростована. Скажімо, якби клас нормально обчислюваних функцій виявився ширше класу рекурсивних функцій, то існувала б нормально обчислювана, але не рекурсивна функція. У силу її нормальної обчислюваності вона була б алгоритмічно обчислювана в інтуїтивному розумінні алгоритму, і припущення про її нерекурсивність спростовувало би тезу Чорча. Але сформульована теорема дійсно справедлива, і таких функцій не існує, що є ще одним непрямим підтвердженням тез Т'юринга, Маркова і Чорча. Зауважимо, що існують ще й інші теорії алгоритмів, і для усіх них також доведено їх рівносильність з розглянутими теоріями.

Вправи

1. Скласти нормальний алгоритм Маркова додавання двох чисел $x + y$.
2. Скласти нормальний алгоритм Маркова множення двох чисел xy.
3. Скласти нормальний алгоритм Маркова копіювання числа x.

4. Скласти нормальний алгоритм Маркова що додає цифру 2 між сусідніми цифрами десяткового запису числа x.
5. Скласти нормальний алгоритм Маркова додавання двох двійкових чисел.
6. Перетворити будь-яке слово в алфавіті $A=(a_0, a_1, a_2, ..., a_n)$ в порожнє слово.
7. Перетворити будь-яке парне число в алфавіті $A = (1)$ в порожнє слово, а непарне — в слово 1.
8. Збільшити довільне число в алфавіті $A = (0, 1, 2, ..., 9)$ на 1.
9. Зменшити довільне число в алфавіті $A = (0, 1, 2, ..., 9)$ на 1.

Розділ 9

АЛГОРИТМІЧНО НЕРОЗВ'ЯЗНІ ПРОБЛЕМИ

Проблема, в якій потрібно знайти єдиний метод (алгоритм) для розв'язання нескінченної серії однотипних одиничних задач називається *алгоритмічною проблемою*. Також такі проблеми називають *масовими проблемами*. Вони виникали і розв'язувалися в різних областях математики протягом усієї її історії. Приклади таких проблем наведено в першому розділі.

На початку минулого століття, як зазначалось, у математиків з'явились сумніви, що деякі масові проблеми не мають загального алгоритму для їх розв'язування. Тому виникла необхідність дати точне означення самому поняттю алгоритму. Ми ознайомилися з декількома способами такого уточнення, і в цьому параграфі приведено приклади алгоритмічно нерозв'язних масових проблем. Як поняття, що уточнює поняття алгоритму, буде використовуватись поняття машини Т'юринга. Почнемо з побудови прикладу функції, що не може бути обчислена на жодній машині Тьюринга.

9.1. ПРИКЛАД ФУНКЦІЇ, ЩО НЕ ОБЧИСЛЮЄТЬСЯ

Наведемо спочатку процес нумерації всіх машин Т'юринга, що використовується при побудові прикладу. Вважаємо, що для позначення внутрішніх станів машин Т'юринга використовуються літери нескінченної послідовності: $q_0, q_1, q_2, ..., q_{rs}$ ..., а для позначення літер зовнішніх алфавітів використовуються літери послідовності: $a_0, a_1, a_2, ..., a_s$ ....

Усі символи скінченних послідовностей виразимо (або, як говорять, закодуємо) словами скінченного стандартного алфавіту $\{a_0, 1, q, П, Л\}$ за такими правилами:

q_0 позначимо (закодуємо) словом q

q_1 позначимо (закодуємо) словом qq

q_2 позначимо (закодуємо) словом qqq

....

q_i позначимо (закодуємо) словом $\underbrace{qq...q}_{i+1}$

....

a_1 позначимо (закодуємо) словом 1

a_2 позначимо (закодуємо) словом 11

a_3 позначимо (закодуємо) словом 111

....

a_j позначимо (закодуємо) словом $\underbrace{11...1}_{j}$

....

Програму машини Тьюринга у стандартному алфавіті можна записати у вигляді слова, застосовуючи таке правило. Спочатку всі команди програми перекладаються мовою стандартного алфавіту, для чого в записах цих команд $q_ia_j \rightarrow q_la_mX$, где $X \in \{С, П, Л\}$, опускається символ «→», а літери q_i, a_j, q_l, a_m заміняються відповідними словами стандартного алфавіту. Потім отримані слова-команди записуються підряд у будь-якому порядку у вигляді єдиного слова.

Зокрема, програма машини Т'юринга, розглянута в прикладі 2.1, має вигляд: $q_1a_0 \rightarrow q_2a_0П$, $q_1a_1 \rightarrow q_2a_1П$, $q_2a_0 \rightarrow q_0a_1С$, $q_2a_1 \rightarrow q_2a_1П$. Опускаємо символ «→», замінюємо літери словами стандартного алфавіту й у результаті одержуємо такі слова в стандартному алфавіті, що кодують відповідні команди: $qqa_0qqqa_0П$, $qq1qq1П$, $qqqa_0q1С$, $qqq1qqq1П$. Виписуємо ці слова підряд і одержуємо слово, що кодує програму даної машини Т'юринга:

$$qqa_0qqqa_0Пqq1qq1Пqqqa_0q1Сqqq1qqq1П.$$

Неважко вказати алгоритм, що дозволяє дізнаватися, чи є слово в стандартному алфавіті програмою деякої машини Т'юринга. Такий алгоритм може, наприклад, полягати в такому. Потрібно аналізувати всі підслова даного слова, вкладені між всіма можливими парами літер із {С, П, Л}. Ці підслова повинні мати таку структуру: спочатку записані кілька літер q, потім a_0 або кілька літер 1, потім знову – кілька літер q, і нарешті знову a_0 або декілька одиниць.

У такий спосіб кожна машина Т'юринга цілком визначається деяким скінченним словом у скінченному стандартному алфавіті. Оскільки множина усіх скінченних слів у скінченному алфавіті зчисленна, то і всіх евентуальних машин Т'юринга (що відрізняються одна від одної по суті своєї роботи) є не більш ніж скінченна кількість.

Пронумеруємо тепер усі машини Т'юринга, для чого всі слова стандартного алфавіту, що представляють собою програми машин Т'юринга, розташуємо у вигляді фіксованої нескінченної послідовності, яку складемо за таким правилом: спочатку виписуються в якій-небудь фіксованій послідовності всі однолітерні слова: $\alpha_0, \alpha_1, ..., \alpha_\xi$, що є програмами машин Т'юринга (множина таких слів скінченна, тому що скінченний стандартний алфавіт, з літер якого будуються слова), потім виписуються всі дволітерні слова $\alpha_{\xi+1},..., \alpha_\eta$, що представляють програми машин Т'юринга (множина таких слів також скінченна, тому що скінченний стандартний алфавіт), потім виписуються трилітерні слова і т.д. Отримаємо послідовність програм $\alpha_0, \alpha_1, \alpha_2, ..., \alpha_n, ...$ усіх мислимих машин Т'юринга. Число k будемо називати *номером машини Т'юринга,* якщо програма цієї машини записується словом α_k.

Розглянемо тепер множину усіх функцій, що задані та приймають значення в множині слів алфавіту $A_1 = \{1\}$. Можна показати, що ця множина незчисленна. З іншого боку, оскільки множина всіх можливих машин Т'юринга, як ми тільки що встановили, перенумерувавши їх, зчисленна, то й множина функцій, обчислюваних за Т'юрингом, також зчисленна. Отже, існують функції, які не обчислювані за Т'юрингом. Вкажемо конкретну функцію, яку не можна обчислити на жодній машині Т'юринга. Згідно тези Т'юринга, це буде означати, що не існує взагалі жодного алгоритму для обчислення значень такої функції.

Розглянемо таку функцію $\varphi(\alpha)$ на словах у алфавіті $A_1 = \{1\}$. Для довільного слова α довжини n в алфавіті $A_1 = \{1\}$ покладемо:

$$\varphi(\alpha) = \begin{cases} \beta_n 1, & \text{якщо слово } \alpha \text{ переробляється машиною Тьюринга} \\ & \text{з номером } n \text{ у слово } \beta_\text{n} \text{ в алфавіті } A_1 = \{1\}, \\ 1, & \text{в іншому випадку.} \end{cases}$$

ТЕОРЕМА 9.1.1. *Функція $\varphi(\alpha)$ не обчислюється за Т'юрингом.*

Доведення. Припустимо протилежне, тобто, що існує машина Т'юринга $\mathfrak{T}$ зі стандартним алфавітом $\{a_0$, 1, q, П, Л), яка обчислює цю функцію. Нехай k — номер цієї машини в щойно описаній нумерації всіх машин Т'юринга. Подивимося, чому дорівнює слово $\varphi\underbrace{(11...1)}_{k}$, що є значенням функції $\varphi(\alpha)$ при $\varphi = \underbrace{11...1}_{k}$. Припустимо, що машина $\mathfrak{T}$ переробляє слово $\underbrace{11...1}_{k}$ в слово β_k в тому ж алфавіті A_1=$\{1\}$. Тоді за означенням обчислюваності функції $\varphi(\alpha)$ на машині $\mathfrak{T}$, це означає, що $\varphi\underbrace{(11...1)}_{k} = \beta_k$. Але з іншого боку, за означенням самої функції $\varphi(\alpha)$, це означає, що $\varphi\underbrace{(11...1)}_{k} = \beta_k 1$. Отримана суперечність доводить, що машини Т'юринга, що обчислює функцію $\varphi(\alpha)$, не існує. Теорему доведено.

На підставі вище доведеної теореми, враховуючи тезу Т'юринга, маємо, що не існує алгоритму для обчислення значень функції $\varphi(\alpha)$. Це означає, що масова проблема знаходження значень функції $\varphi(\alpha)$, для всіх можливих значень аргументу алгоритмічно нерозв'язна.

9.2. ПРОБЛЕМА РОЗПІЗНАВАННЯ САМОЗАСТОСОВНОСТІ

Це ще один приклад нерозв'язної алгоритмічної проблеми. Припустимо, що на стрічці машини Т'юринга записана її власна функціональна схема в алфавіті машини. Якщо машина застосовна до такої конфігурації, то будемо називати її самозастосовною, у протилежному випадку — несамозастосовною. Виникає масова проблема розпізнавання самозастосовних машин Т'юринга, яка полягає в тому, що за заданою функціональною схемою (програмою) машини Т'юринга встановити, до якого класу відноситься машина: до класу самозастосовних машин або до класу несамозастосовних.

ТЕОРЕМА 9.2.1. *Проблема розпізнавання самозастосовних машин Т'юринга алгоритмічно нерозв'язна.*

Доведення. Припустимо протилежне, тобто що алгоритм для такого розпізнавання існує. Тоді, згідно тези Т'юринга, існує машина Т'юринга, що реалізує даний алгоритм. Нехай $\mathfrak{T}$ — така машина. На її стрічку заноситься відповідним чином закодована програма тієї або іншої машини Т'юринга. При цьому, якщо машина самозастосовна, то занесене слово переробляється машиною $\mathfrak{T}$ у якийсь символ σ (що має зміст позитивної відповіді на поставлене питання про самозастосовність). Якщо ж машина несамозастосовна, то занесене на стрічку слово, що кодує її програму, переробляється машиною $\mathfrak{T}$ у якийсь символ τ (що має зміст негативної відповіді на поставлене питання).

Розглянемо тепер таку машину Т'юринга $\mathfrak{T}_1$, що як і раніше переробляє несамозастосовні коди в τ, а до самозастосовних кодів машина $\mathfrak{T}_1$ вже незастосовна. Таку машину отримаємо з машини $\mathfrak{T}$, якщо злегка змінити її програму в такий спосіб: після появи символу σ замість зупинки машина повинна необмежено його повторювати.

Отже, $\mathfrak{T}_1$ застосовується до будь-якого несамозастосовного коду (виробляючи символ τ) і незастосовна до самозастосовних кодів. Це приводить до суперечності, тому що така машина не може бути ні самозастосовною, ні несамозастосовною. Справді, якщо машина $\mathfrak{T}_1$ самозастосовна, то вона незастосовна до свого коду. Виходить, машина несамозастосовна. Маємо суперечність. З іншого боку, якщо машина $\mathfrak{T}_1$ несамозастосовна, то її код мусить перероблятися самою машиною $\mathfrak{T}_1$ у символ τ. Отже, $\mathfrak{T}_1$ застосовна до власного коду, тобто самозастосовна. Знову суперечність. Вона і доводить теорему.

Згідно доведеної теореми встановлюється алгоритмічна нерозв'язність і деяких інших масових проблем, що виникають у теорії машин Т'юринга. Наприклад, *алгоритмічно нерозв'язна проблема розпізнавання застосовності для машин Т'юринга*. Проблема полягає в тому, що задано функціональну схему (програму) деякої машини Т'юринга і конфігурація в ній: потрібно з'ясувати, застосовна машина до даної конфігурації чи ні. Якби існував алгоритм для розв'язання цієї проблеми, то з його допомогою можна було б дізнаватися, чи застосовна машина до слова, що кодує її власну програму, тобто чи самозастосовна вона. Але, на підставі попередньої теореми, відомо, що такого алгоритму не існує.

9.3. ІНШІ ПРИКЛАДИ АЛГОРИТМІЧНОЇ НЕРОЗВ'ЯЗНОСТІ

Як відомо, не існує алгоритму, що дозволяє для кожної формули логіки предикатів визначити, чи буде формула виконуваною або загальнозначущою. Це означає, що *загальні проблеми вирішення для загальнозначущості та виконуваності формул алгебри предикатів алгоритмічно нерозв'язні.*

Найбільш відомою алгоритмічною проблемою математики була *десята проблема Гільберта,* поставлена ним у числі інших у 1901 році на Міжнародному математичному конгресі в Парижі. Потрібно було знайти алгоритм, що визначає для будь-якого діофантового рівняння, чи має воно цілочисельний розв'язок. Діофантове рівняння — це рівняння виду $F(x, y, ..., z) = 0$, де $F(x, y, ..., z)$ — многочлен з цілими коефіцієнтами. У загальному випадку ця проблема довго залишалася невирішеною і тільки в 1970 році російський математик Ю.В. Матиясевич довів її нерозв'язність,

Відзначимо ще раз на закінчення, що алгоритмічна нерозв'язність означає лише відсутність єдиного способу для розв'язку всіх одиничних задач даної нескінченної серії, у той час як кожна індивідуальна задача серії цілком може бути розв’язана своїм індивідуальним способом. Так, не зважаючи на відсутність єдиного алгоритму, що дозволяє для кожної формули логіки предикатів визначити, чи є вона виконувана або загальнозначуща, існує відповідь на це питання стосовно до конкретних формул. Аналогічна ситуація з діофантовими рівняннями. Наприклад, для випадку алгебраїчного рівняння з цілими коефіцієнтами

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

добре відомо, що всі його цілі корені потрібно шукати серед дільників вільного члена a_0.

Існують й інші алгоритмічні проблеми, щодо яких встановлена їх нерозв'язність. Серед них ряд алгебраїчних проблем, що приводять до різних варіантів проблеми слів.

ДОДАТКИ

ЛОГІЧНІ ОПЕРАЦІЇ НАД ВИСЛОВЛЕННЯМИ

Назва	Позначення	Тип	Інші позначення
заперечення	$\rceil$	*унарний*	$\overline{\ }$, *not*, *ні*
кон'юнкція	$\wedge$	*бінарний*	*&*, *and*, *і*
диз'юнкція	$\vee$	*бінарний*	*or*, *або*
імплікація	$\Rightarrow$	*бінарний*	$\supset$, $\rightarrow$
еквіваленція	$\Leftrightarrow$	*бінарний*	$\sim$, $\leftrightarrow$

ТАБЛИЦІ ЕЛЕМЕНТАРНИХ ФУНКЦІЙ $\rceil a$, $a \wedge b$, $a \vee b$, $a \Rightarrow b$, $a \Leftrightarrow b$

a	b	$\rceil a$	$a \wedge b$	$a \vee b$	$a \Rightarrow b$	$a \Leftrightarrow b$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

ДЕЯКІ НАЙВАЖЛИВІШІ ТАВТОЛОГІЇ

1. $a \Leftrightarrow \rceil\rceil a$ — закон подвійного заперечення;
2. $a \vee \rceil a$ — закон виключеного третього;
3. $\rceil(a \wedge \rceil a)$ — закон виключення суперечності;
4. $a \Leftrightarrow a$ — закон тотожності;
5. $a \wedge a \Leftrightarrow a$, $a \vee a \Leftrightarrow a$ — закони ідемпотентності;
6. $a \wedge b \Rightarrow a$, $a \wedge b \Rightarrow b$ — закон спрощення;
7. $a \Rightarrow a \vee b$, $b \Rightarrow a \vee b$ — закон спрощення;
8. $(a \Rightarrow b) \Leftrightarrow (\rceil b \Rightarrow \rceil a)$ — закон контрапозиції;
9. $(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$ — закон силогізму;
10. $(a \Leftrightarrow b) \Leftrightarrow (\rceil a \Leftrightarrow \rceil b)$ — закон протилежності;
11. $a \Rightarrow (b \Rightarrow a)$ — істина з чого завгодно;
12. $\rceil a \Rightarrow (a \Rightarrow b)$ — з хибного що завгодно;

13. $a \wedge (a \Rightarrow b) \Rightarrow b$ — закон modus ponens;
14. $(a \Rightarrow b) \wedge \neg b \Rightarrow \neg a$ — закон modus tollens;
15. $\neg(a \wedge b) \Leftrightarrow \neg a \vee \neg b$ — закон де-Моргана;
16. $\neg(a \vee b) \Leftrightarrow \neg a \wedge \neg b$ — закон де-Моргана;
17. $a \wedge (a \vee b) \Leftrightarrow a$ — закон поглинання;
18. $a \vee a \wedge b \Leftrightarrow a$ — закон поглинання;
19. $a \Leftrightarrow (b \Rightarrow a \wedge b)$.

ОСНОВНІ РІВНОСИЛЬНІ ФОРМУЛИ

1. $a \wedge b \equiv b \wedge a$ комутативний закон;
2. $a \vee b \equiv b \vee a$ комутативний закон;
3. $a \wedge (b \wedge c) \equiv (a \wedge b) \wedge c$ асоціативний закон;
4. $a \vee (b \vee c) \equiv (a \vee b) \vee c$ асоціативний закон;
5. $a \wedge (b \vee c) \equiv a \wedge b \vee a \wedge c$ перший дистрибутивний закон;
6. $a \vee b \wedge c \equiv (a \vee b) \wedge (a \vee c)$ другий дистрибутивний закон;
7. $a \wedge a \equiv a$ закон ідемпотентності;
8. $a \vee a \equiv a$ закон ідемпотентності;
9. $a \vee a \wedge b \equiv a$ закон поглинання;
10. $a \wedge (a \vee b) \equiv a$ закон поглинання;
11. $\neg(a \wedge b) \equiv \neg a \vee \neg b$ закон де-Моргана;
12. $\neg(a \vee b) \equiv \neg a \wedge \neg b$ закон де-Моргана;
13. $a \Rightarrow b \equiv \neg b \Rightarrow \neg a$ закон контрапозиції;
14. $a \Rightarrow b \equiv \neg a \vee b$;
15. $a \Leftrightarrow b \equiv (a \Rightarrow b) \wedge (b \Rightarrow a)$;
16. $\neg\neg a \equiv a$ закон подвійного заперечення;
17. $a \vee \neg a = 1$ закон виключеного третього;
18. $a \wedge \neg a \equiv 0$ закон суперечності.

Для довільної формули A мають місце закони сталих:

19. $A \wedge 0 \equiv 0$;
20. $A \wedge 1 \equiv A$;
21. $A \vee 0 \equiv A$;
22. $A \vee 1 \equiv 1$.

АКСІОМИ ЧИСЛЕННЯ ВИСЛОВЛЕНЬ

I

A1. $a \Rightarrow (b \Rightarrow a)$;

A2. $(a \Rightarrow (b \Rightarrow c)) \Rightarrow ((a \Rightarrow b) \Rightarrow (a \Rightarrow c))$;

II

A3. $a \wedge b \Rightarrow a$;

A4. $a \wedge b \Rightarrow b$;

A5. $(a \Rightarrow b) \Rightarrow ((a \Rightarrow c) \Rightarrow (a \Rightarrow b \wedge c))$;

III

A6. $a \Rightarrow a \vee b$;

A7. $b \Rightarrow a \vee b$;

A8. $(a \Rightarrow c) \Rightarrow ((b \Rightarrow c) \Rightarrow (a \vee b \Rightarrow c))$;

IV

A9. $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$;

A10. $a \Rightarrow \neg\neg a$;

A11. $\neg\neg a \Rightarrow a$.

ПОХІДНІ ПРАВИЛА ВИВЕДЕННЯ

1. $A \Rightarrow B, B \Rightarrow C \vdash A \Rightarrow C$ — *правило силогізму ПС;*
2. $A \Rightarrow (B \Rightarrow C) \vdash B \Rightarrow (A \Rightarrow C)$ — *правило перестановки посилок ППП;*
3. $A \Rightarrow (B \Rightarrow C) \vdash A \wedge B \Rightarrow C$ — *правило з’єднання посилок ПЗП;*
4. $A \wedge B \Rightarrow C \vdash A \Rightarrow (B \Rightarrow C)$ — *правило роз’єднання посилок ПРП;*
5. $A, B \vdash A \wedge B$ — *правило введення кон’юнкції ВвК;*
6. $A \vdash A \vee B, B \vdash A \vee B$ — *правила введення диз’юнкції ВвД;*
7. $A \wedge B \vdash A, A \wedge B \vdash B$ — *правила вилучення кон’юнкції ВК;*
8. $\neg\neg A \vdash A$ — *правило вилучення заперечення ВЗ;*
9. $A \Rightarrow B, A \Rightarrow C \vdash A \Rightarrow B \wedge C$ — *правило композиції ПК;*
10. $A \Rightarrow B \vdash \neg B \Rightarrow \neg A$ — *правило контрапозиції КП;*

11. $\dfrac{\Gamma, A \vdash C, \quad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}$ — *правило вилучення диз'юнкції ВД;*

12. $\dfrac{\Gamma, A \vdash B; \quad \Gamma, A \vdash \neg B}{\Gamma, \vdash \neg A}$ — *правило введення заперечення або приведення до абсурду ПА (Reductio ad absurdum RA).*

ЛОГІЧНО ЗАГАЛЬНОЗНАЧУЩІ ФОРМУЛИ ЛОГІКИ ПРЕДИКАТІВ

1. $\forall x\, P(x) \Rightarrow P(y)$;
2. $P(y) \Rightarrow \exists x\, P(x)$;
3. $\neg \forall x P(x) \Leftrightarrow \exists x\, \neg P(x)$
4. $\neg \exists x\, P(x) \Leftrightarrow \forall x \neg (P(x)$

$\left.\begin{array}{l} 3 \\ 4 \end{array}\right\}$ Закони де Моргана для кванторів;

5. $\forall x(P(x) \wedge Q(x)) \Leftrightarrow \forall x P(x) \wedge \forall x Q(x)$;
6. $\exists x(P(x) \vee Q(x)) \Leftrightarrow \exists x\, P(x) \vee \exists x\, Q(x)$;
7. $\forall x\, P(x) \vee \forall x\, Q(x) \Rightarrow \forall x(P(x) \vee Q(x))$;
8. $\exists x(P(x) \wedge Q(x)) \Rightarrow \exists x P(x) \wedge \exists x Q(x)$;
9. $\forall x(A(x) \wedge B) \Leftrightarrow \forall x A(x) \wedge B$;
10. $\forall x(A(x) \vee B) \Leftrightarrow \forall x A(x) \vee B$;
11. $\exists x(A(x) \wedge B) \Leftrightarrow \exists x A(x) \wedge B$;
12. $\exists x(A(x) \vee B) \Leftrightarrow \exists x A(x) \vee B$;
13. $\forall x(A(x) \Rightarrow B) \Leftrightarrow \exists x\, A(x) \Rightarrow B$;
14. $\exists x(A(x) \Rightarrow B) \Leftrightarrow \forall x\, A(x) \Rightarrow B$;
15. $\forall x(B \Rightarrow A(x)) \Leftrightarrow B \Rightarrow \forall x A(x)$;
16. $\exists x(B \Rightarrow A(x)) \Leftrightarrow B \Rightarrow \exists x A(x)$;
17. $\forall x\, \forall y\, A(x, y) \Leftrightarrow \forall y\, \forall x A(x, y)$;
18. $\exists x\, \exists y\, A(x, y) \Leftrightarrow \exists y\, \exists x\, A(x, y)$;
19. $\exists x\, \forall y\, A(x, y) \Rightarrow \forall y\, \exists x\, A(x, y)$.

ЛІТЕРАТУРА

Основна

1. Ершов Ю.Л., Палютин Е.А. Математическая логика. Москва: Наука, 1979. 320 с.
2. Игошин В.И. Математическая логика и теория алгоритмов. Саратов: Изд-во Сарат. ун-та, 1991. 256 с.
3. Клини С.К. Математическая логика. Москва: Мир, 1973. 420 с.
4. Мендельсон Э. Введение в математическую логику. Москва: Наука,1984. 320 с.
5. Марков А.А., Нагорный Н.М. Теория алгорифмов. Москва: Наука, 1984. 432 с.
6. Новиков П.С. Элементы математической логики. Москва: Наука, 1973. 400 с.

Додаткова

1. Авдеюк П.І. Елементи математичної логіки. Логіка висловлень. Кам'янець-Подільський: Кам'янець-Подільський державний педагогічний інститут, 1996. 60 с.
2. Авдеюк П.І. Елементи математичної логіки. Логіка предикатів. Кам'янець-Подільський: Кам'янець-Подільський державний педагогічний університет, 2000. 60 с.
3. Авдеюк П.І. Елементи математичної логіки. Кам'янець-Подільський: Кам'янець-Подільський державний педагогічний університет, 2001. 100 с.
4. Авдеюк П.І. Елементи теорії алгоритмів. Кам'янець-Подільський: Видавець ПП Зволейко Д.Г., 2011. 32 с.
5. Алферова З.Б. Теория алгоритмов. Москва: Статистика, 1973. 154 с.
6. Игошин В.И. Задачник-практикум по математической логике. Москва: Просвещение, 1986. 159 с.
7. Латотин Л.А., Макаренков Ю.А., Николаева В.В., Столяр А.А. Математическая логика. Минск: Вышейшая школа, 1991. 270 с.
8. Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов. Москва: Наука, 1975. 240 с.

9. Успенский В.А., Семенов А.Л. Теория алгоритмов: основные открытия и приложения. Москва: Наука, ГРФМЛ, 1987. 136 с.
10. Хромой Я.В. Збірник вправ і задач з математичної логіки. Київ: Вища школа, 1978. 160 с.
11. Хромой Я.В. Математична логіка. Київ: Вища школа, 1983. 208 с.

Міністерство освіти і науки України
Кам'янець-Подільський національний університет
імені Івана Огієнка

НАВЧАЛЬНЕ ВИДАННЯ

АВДЕЮК Павло Іванович,

кандидат фізико-математичних наук, доцент, доцент кафедри математики Кам'янець-Подільського національного університету імені Івана Огієнка

ЗЕЛЕНСЬКИЙ Олексій Віталійович,

кандидат фізико-математичних наук, доцент кафедри математики Кам'янець-Подільського національного університету імені Івана Огієнка

ЕЛЕМЕНТИ МАТЕМАТИЧНОЇ ЛОГІКИ ТА ТЕОРІЇ АЛГОРИТМІВ

Навчально-методичний посібник

Підписано до друку 14.05.2019 р. Гарнітура Петербург.
Папір офсетний. Друк різографічний.
Формат 60x84/16. Умовн. друк. арк. 9,3. Обл.-вид. арк. 8,2.
Тираж 50. Зам. № 855.

Кам'янець-Подільський національний університет
імені Івана Огієнка, вул. Огієнка, 61. Кам'янець-Подільський, 32300.
Свідоцтво про внесення до державного реєстру
суб'єктів видавничої справи серії ДК № 3382 від 05.02.2009 р.

Надруковано в Кам'янець-Подільському
національному університеті імені Івана Огієнка,
вул. Огієнка, 61. Кам'янець-Подільський, 32300.