**Вінницький державний педагогічний університет
імені Михайла Коцюбинського**

В. С. Трохименко

КОНСПЕКТ ЛЕКЦІЙ З МАТЕМАТИЧНОЇ ЛОГІКИ ТА ТЕОРІЇ АЛГОРИТМІВ

Вінниця – 2007

Трохименко Валентин Степанович, *професор, кандидат фіз.-мат. наук.*

Курс математичної логіки і теорії алгоритмів в педагогічних університетах має на меті ознайомити майбутніх учителів математики з основами цієї науки, оскільки вона посідає важливе місце в професійній підготовці вчителів математики. Вона сприяє вихованню культури логічного мислення, кращому розумінню структурно-логічної схеми шкільного курсу математики, глибшому проникненню в суть процесу доведення теорем та встановлення зв'язків між ними. Символіка та мова математичної логіки дають змогу стисло і точно описувати означення математичних понять, формування теорем та їх доведень. Математична логіка дає вчителю нові засоби для формування в учнів навичок точного мислення.

Конспект лекцій відповідає діючій програмі з математичної логіки і теорії алгоритмів для математичних спеціальностей педагогічних університетів. Ним можуть користуватись не тільки студенти стаціонарного відділення, але й заочного, та особи, які вивчають цей курс самостійно.

Зміст

Вступ

Математична логіка належить до такого напряму в математиці, який останнім часом особливо інтенсивно розвивається. Це пояснюється зростаючим у наш час значенням математичної логіки в таких галузях, як проектування обчислювальних машин і автоматичних систем, програмування та кібернетики.

Курс математичної логіки в педагогічних університетах має на меті ознайомити майбутніх учителів математики з основами цієї науки. Конспект лекцій охоплює найголовніші питання математичної логіки.

Перший розділ — "Логіка висловлень" — викладений змістовно, вводить студента в коло основних понять: висловлення, логічні операції і функції; ознайомлює з символікою та апаратом алгебри логіки, необхідних для подальшого вивчення курсу.

Зважаючи на практичне значення цього розділу при конструюванні систем керування та лічильно-цифрових пристроїв, до нього включено попереднє ознайомлення з релейно-контактними схемами.

Другий розділ — "Числення висловлень" — будується на формально аксіоматичній основі. Важливість цього розділу визначається тим, що він входить як частина в більш широкі логічні теорії. Завдяки своїй простоті, порівняно з іншими аксіоматичними теоріями числення, числення висловлювань дає змогу без великої затрати часу проілюструвати на ньому багато питань математичної логіки: формальну довідність, несуперечливість, повноту та ін.

У третьому розділі — "Логіка предикатів" — логіка предикатів викладається змістовно. В ньому студенти ознайомляться з апаратом логіки предикатів і використанням його при формуванні математичних тверджень.

У четвертому розділі — "Математичні теорії першого порядку" — передбачено ознайомити студентів з мовою першого порядку, розглянути логічні та спеціальні аксіоми, навести приклади формалізованих теорій першого порядку. Розглядаються питання несуперечності, повноти та незалежності аксіом числення предикатів. Студенти знайомляться з аксіомами формальної арифметики та формулюванням теореми Геделя про неповноту. Проблема вирішення, що включена до цього розділу, пов'язана з багатьма важливими питаннями сучасної математики.

П'ятий розділ — "Елементи теорії алгоритмів" — має на меті з'ясування поняття про алгоритм спочатку на інтуїтивній основі, з розглядом прикладів з різних розділів математики. З багатьох еквівалентних між собою уточнень алгоритму розглядається визначення алгоритму у формі машини Тьюрінга.

1 Логіка висловлень

1.1 Логічні операції над висловленнями

Предмет математичної логіки. Логічні операції над висловленнями. Формули і таблиці істинності. Булеві функції та їх число. Тавтології.

1. Згідно одного з розповсюджених визначень під *логікою* розуміють науку *про методи міркувань*. Причому вивчаючи ці методи, логіка цікавиться в першу чергу формою, а не змістом доказів у тому чи іншому міркуванні. Розглянемо, наприклад, такі два міркування:

- Всі люди смертні. Сократ — людина. Отже, Сократ — смертний.
- Всі кролі люблять моркву. Себаст'ян — кролик. Отже, Себяст'ян любить моркву.

Обидва ці міркування мають одну і ту ж форму: "Всі A суть B. S є A. Отже, S є B". Істинність або хибність окремих посилок або висновків не цікавить логіка. Він бажає лише знати чи випливає істинність висновку з істинності посилок.

Систематична формалізація і каталогізація правильних способів міркувань — одна з головних задач логіка. Якщо при цьому логік застосовує математичний апарат, і його дослідження присвячені в першу чергу вивченню математичних міркувань, то предмет його діяльності може бути названий *математичною логікою*. Головна мета математичної логіки — дати точне визначення поняттю *"математичне доведення"*.

Ще в давні часи, приблизно 2.5 тисячі років тому назад, в Індії, Китаї та Греції мислителі й філософи почали систематично вивчати загальні форми логічних умовиводів. Основний вплив вплив на розвиток логіки зробила старогрецька формальна логіка, розвинута в основному Аристотелем (384 – 322 р. до н.е.) в його працях "Органон", "Аналітика", "Топіка", "Категорики"та ін. Ідею побудови математичної логіки вперше чітко сформулював Лейбніц (1646 – 1716), однак його праці містять лише програму побудови алгебри логіки, оскільки в ті часи не було необхідності в побудові математичної логіки. Ця потреба виникла в середині XIX ст., і першими творцями математичної логіки вважають англійських математиків Джорджа Буля та Августа де Моргана. Скоро новий напрямок в математиці привернув до себе увагу спеціалістів: в Німеччині — Шредера і Фреге; Франції — Кутюра і Бутру; Італії — Пеано; США — Пірса; Росії — професора Казанського університету П. С. Порецького (1846 – 1907).

Особливо швидкий розвиток математичної логіки спостерігається на початку XX ст. і продовжується в наш час у зв'язку з дослідженнями *основ математики*. Це пов'язано з появою на початку століття парадоксів. Розглянемо, наприклад, відомий парадокс Рассела (1902 р.). Отже, нехай A є множина всіх таких множин X, які не є своїм елементом, тобто $A = \{X \mid X \notin X\}$. Згідно означення, якщо A є своїм елементом, то $A \notin A$; якщо ж A не є своїм елементом, то $A \in A$. Отже, одночасно виконуються $A \in A$ і $A \notin A$. Існують й інші парадокси (див. [17]). Всі парадокси є дійсними в тому розумінні, що вони не містять логічних вад. Спроби математиків позбутися парадоксів в теорії множин призвели їх до створення цілого ряду аксіоматичних теорій множин, а це вимагало подальшого розвитку математичної логіки.

Новий стимул в розвитку математичної логіки пов'язаний із створенням електронних обчислювальних машин, з розвитком нової науки *кібернетики*. Тут результати логіки знайшли чисто технічні застосування.

В наш час ідеї математичної логіки використовуються не тільки в математиці або кібернетиці, але й лінгвістиці, біології, радіотехніці та ін.

2. Одним з основних понять математичної логіки є поняття *висловлення*, під яким розуміється стверджувальне речення, відносно якого ми можемо запитати *істинне* воно чи *хибне*. Наприклад, речення "Париж — столиця Франції", "8 є число непарне", "всі дійсні числа задовольняють нерівність $x^2 < 9$" — приклади висловлень. Висловлення, які є простими реченнями називаються *простими*, а всі останні — *складними*. В математичній логіці складні висловлення утворюються з простих за допомогою часток або сполучників. Найбільш використовуваними є частка *"не"* і сполучники *"і", "або", "якщо. . . , то. . . ", "якщо. . . , то. . . і навпаки"*, які називаються *логічними операціями* (або *логічними зв'язками*).

В подальшому висловлення ми будемо позначати латинськими літерами A, B, $C, \ldots$; p, q, r, $s, \ldots$, при цьому значення *"істина"* будемо позначати символом "1", а *"хиба"* — "0". Змінні величини, які приймають два значення *"істина"* і *"хиба"*, ми будемо називати *логічними змінними*. Істинностне значення логічної змінної A будемо позначати через $|A|$.

Для позначення логічних операцій введемо такі позначення: $\sim A$ або $\overline{A}$ читається *"не A"* або *"невірно, що A"*; AB або $A \wedge B$ читається *"A і B"*;[1] $A \vee B$ читається *A або B*; $A \longrightarrow B$ читається *"якщо A, то B"*, *"з A випливає B"*, *"A достатня умова для B"*, *"B необхідна умова A"*; $A \longleftrightarrow B$ читається *"якщо A, то B і навпаки", "A еквівалентне B", "A необхідна і достатня умова для B", "B тоді і тільки тоді коли A"*.[2] Логічні операції мають такі назви: $\sim$ — заперечення, $\wedge$ — кон'юнкція, $\vee$ — диз'юнкція, $\longrightarrow$ — імплікація, $\longleftrightarrow$ — еквівалентність. Встановлюється порядок виконання логічних операцій, а саме, в такому порядку: $\sim$, $\wedge$, $\vee$, $\longrightarrow$, $\longleftrightarrow$. Якщо вказаний порядок необхідно змінити, то використовують дужки. Логічні операції визначаються такими таблицями істинності:

A	$\sim A$
0	1
1	0

A	B	$A \wedge B$	$A \vee B$	$A \longrightarrow B$	$A \longleftrightarrow B$
0	0	0	0	1	1
1	0	0	1	0	0
0	1	0	1	1	0
1	1	1	1	1	1

Відмітимо, що кон'юнкцію часто називають *логічним множенням*, а диз'юнкцію — *логічним додаванням*.

3. Нехай $p, q, r, s, t, w, \ldots$ є логічні змінні. За допомогою логічних операцій з них ми можемо утворити вирази виду $\sim p$, $p \wedge q$, $p \vee q$, $p \longrightarrow q$, $p \longleftrightarrow q$ і т.д. З отриманих виразів ми далі можемо утворювати більш складні вирази, наприклад, $p \wedge q \longrightarrow s \vee t$, $(u \longleftrightarrow p) \longrightarrow\sim q$, $\sim (p \vee q) \longrightarrow r \wedge s$ і т.д. Подібні вирази ми в подальшому будемо називати *формулами логіки висловлень*.[3] Для кожної конкретної формули, знаючи істинностні значення її логічних змінних, ми можемо обчислити істинностні значення всієї формули. Нехай, скажімо, у формулі $\sim (p \vee q) \longrightarrow r \wedge s$ ми маємо $|p| = 1$, $|q| = 0$,

[1] В деяких книгах зустрічається також позначення $A\&B$.

[2] В останніх чотирьох реченнях літери A і B можна поміняти місцями.

[3] Чітке означення формул можна знайти в книгах [1], [2], [3], [10].

$|r| = 1$, $|s| = 1$, тоді формула приймає значення $\sim (1 \vee 0) \longrightarrow 1 \wedge 1$, тобто $\sim 1 \longrightarrow 1$ або $0 \longrightarrow 1$, що означає 1. Отже, дана формула при даних істинностних значеннях змінних приймає значенні "істина".

Кожна формула логіки висловлень на множині $\{0, 1\}$ визначає функцію, яка приймає значення на цій же множині. Ця функція може бути задана табличним способом. В математичній логіці такі таблиці називають *таблицями істинності*. Наприклад, формула $\sim p \vee q \longrightarrow s$ визначає функцію від трьох аргументів, яка має таку таблицю істинності:

p	q	s	$\sim p$	$\sim p \vee q$	$\sim p \vee q \longrightarrow s$
0	0	0	1	1	0
1	0	0	0	0	1
0	1	0	1	1	0
1	1	0	0	1	0
0	0	1	1	1	1
1	0	1	0	0	1
0	1	1	1	1	1
1	1	1	0	1	1

Неважко бачити, що число рядків у таблиці істинності для формули, яка містить n логічних змінних, дорівнює 2^n.

4. Функції, які визначені на двохелементній множині $\{0, 1\}$, називаються *булевими*. Вони відіграють в математичній логіці досить важливу роль, тому приділяємо деяку увагу до вивчення таких функцій. Отже, нехай $f(x_1, x_2, \ldots, x_n)$ є булева функція від аргументів $x_1, x_2, \ldots, x_n$. Очевидно, що область визначення такої функції є множина всіх упорядкованих n-ок, компоненти яких є 0 або 1, тобто множина $\{(0, 0, \ldots, 0), (1, 0, \ldots, 0), \ldots, (1, 1, \ldots, 1)\}$. Такі n-ки називають *двійковими наборами*. Вище зазначалося, що число n-вимірних двійкових наборів дорівнює 2^n. *А скільки існує різних булевих функцій від n аргументів?* Відповідь на це питання дає наступна теорема:

Теорема 1. *Число m всіх булевих функцій від n аргументів дорівнює 2^{2^n}, тобто*

$$m = 2^{2^n} \tag{1.1.1}$$

Доведення стає очевидним, якщо скористатись такою таблицею:

x_1	x_2	$\cdots$	x_{n-1}	x_n	f_1	f_2	f_3	$\cdots$	f_m
0	0	$\ldots$	0	0	0	1	0	$\ldots$	1
0	0	$\ldots$	0	1	0	0	1	$\ldots$	1
0	0	$\ldots$	1	0	0	0	0	$\ldots$	1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
1	1	$\ldots$	1	1	0	0	0	$\ldots$	1

В даній таблиці є 2^n рядків по числу всіх двійкових наборів розмірності n. Кожний стовпчик f_i є двійковим набором розмірності 2^n, тому число всіх таких наборів, тобто число всіх функцій m, дорівнює 2^{2^n}. Теорема доведена. □

Отже, булевих функцій від одного аргументу всього 4:

x	0	$\sim x$	x	1
0	0	1	0	1
1	0	0	1	1

Від двох аргументів булевих функцій, згідно формули (1.1.1), буде 16:

x	0	1	0	1	Назва булевих
y	0	0	1	1	функцій
0	0	0	0	0	Константа 0
$x \downarrow y$	1	0	0	0	Стрілка Пірса
$x \not\longrightarrow y$	0	1	0	0	Заперечення імплікації
$\sim y$	1	1	0	0	Заперечення y
$x \not\longleftarrow y$	0	0	1	0	Заперечення оберненої імплікації
$\sim x$	1	0	1	0	Заперечення x
$x \oplus y$	0	1	1	0	Сума по модулю 2
$x \mid y$	1	1	1	0	Штрих Шеффера
$x \wedge y$	0	0	0	1	Кон'юнкція
$x \longleftrightarrow y$	1	0	0	1	Еквівалентність
x	0	1	0	1	x
$x \longleftarrow y$	1	1	0	1	Обернена імплікація
y	0	0	1	1	y
$x \longrightarrow y$	1	0	1	1	Імплікація
$x \vee y$	0	1	1	1	Диз'юнкція
1	1	1	1	1	Константа 1

Булевих функцій від трьох аргументів — 256, від 4-х — 65536, від 5-ти — $\approx$ 4 млрд. Звернемо увагу на дві важливі функції:

$$x \mid y \stackrel{df}{=} \sim (x \wedge y) \text{ — штрих Шеффера,}$$

$$x \downarrow y \stackrel{df}{=} \sim (x \vee y) \text{ — стрілка Пірса.}$$

Дані функції володіють деякими "хорошими"властивостями, про які піде мова згодом.

5. Булева функція називається *тотожно істиною* (або константою 1) відповідно *тотожно хибною* (або константою 0), якщо для довільних значень своїх аргументів вона приймає значення 1 відповідно 0. Формула, яка визначає тотожно істину булеву функцію називається *тавтологією*, відповідно, тотожно хибну — *протиріччям*. Очевидно, что заперечення протиріччя є тавтологія і навпаки. тавтології називають також *законами логіки*. Якщо формула $\mathfrak{A}$ є тавтологією, то цей факт позначається як $\models \mathfrak{A}$. Символ $\models$ називається *подвійним турнікетом*.

Наведемо приклад тавтології. Покажемо, що формула $(p \longrightarrow q) \longrightarrow (p \vee r \longrightarrow q \vee r)$ є тавтологія. Для цього складемо таблицю істинності цієї формули:

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

Відмітимо, що тавтології задовольняють такі дві теореми:

Теорема 2. *Для довільних формул A і B, якщо формули A і $A \longrightarrow B$ є тавтології, то B — тавтологія.*

Доведення теореми очевидне, оскільки воно випливає з означення імплікації. Коротко формулювання теореми 2 записується так:" *Якщо* $\models A$ *і* $\models A \longrightarrow B$, *то* $\models B$."

Теорема 3. *Якщо A є тавтологія, що містить логічні змінні $x_1, x_2, \ldots, x_n$, і формула B отримується з A підстановкою в неї формул $C_1, C_2, \ldots, C_n$ замість $x_1, x_2, \ldots, x_n$, то B є також тавтологія.*

Доведення. Нехай логічні змінні формули B приймають довільно деякі значення. Припустимо, що при цьому формула B приймає значення $|B|$. Оскільки формула B є результат підстановки $S^{x_1,\ldots,x_n}_{C_1,\ldots,C_n}(A)$, то $|B| = S^{x_1,\ldots,x_n}_{|C_1|,\ldots,|C_n|}(A) = 1$, оскільки A — тавтологія. В силу довільності вибору значень для змінних формули B, робимо висновок, що і B — тавтологія. Теорема доведена. □

Коротко формулювання теореми 3 записується так: "Якщо $\models A$, то $\models S^{x_1,\ldots,x_n}_{C_1,\ldots,C_n}(A)$."

СПИСОК ОСНОВНИХ ТАВТОЛОГІЙ ЛОГІКИ ВИСЛОВЛЕНЬ

1. $A \wedge (A \longrightarrow B) \longrightarrow B$,
2. $\sim B \wedge (A \longrightarrow B) \longrightarrow \sim A$,
3. $\sim A \wedge (A \vee B) \longrightarrow B$,
4. $A \longrightarrow (B \longrightarrow A \wedge B)$,
5. $A \wedge B \longrightarrow A$,
6. $A \longrightarrow A \vee B$,
7. $(A \longrightarrow B) \wedge (B \longrightarrow C) \longrightarrow (A \longrightarrow C)$,
8. $(A \wedge B \longrightarrow C) \longrightarrow (A \longrightarrow (B \longrightarrow C))$,
9. $(A \longrightarrow (B \longrightarrow C)) \longrightarrow (A \wedge B \longrightarrow C)$,
10. $(A \longrightarrow B \wedge \sim B) \longrightarrow \sim A$,
11. $(A \longrightarrow B) \longrightarrow (A \vee C \longrightarrow B \vee C)$,

12. $(A \longrightarrow B) \longrightarrow (A \wedge C \longrightarrow B \wedge C)$,
13. $(A \longrightarrow B) \longrightarrow ((B \longrightarrow C) \longrightarrow (A \longrightarrow C))$,
14. $(A \longleftrightarrow B) \wedge (B \longleftrightarrow C) \longrightarrow (A \longleftrightarrow C)$,
15. $A \longleftrightarrow A$,
16. $\sim\sim A \longleftrightarrow A$,
17. $(A \longleftrightarrow B) \longleftrightarrow (B \longleftrightarrow A)$,
18. $(A \longrightarrow B) \wedge (C \longrightarrow B) \longleftrightarrow (A \vee C \longrightarrow B)$,
19. $(A \longrightarrow B) \wedge (A \longrightarrow C) \longleftrightarrow (A \longrightarrow B \wedge C)$,
20. $(A \longrightarrow B) \longleftrightarrow (\sim B \longrightarrow \sim A)$,
21. $A \vee B \longleftrightarrow B \vee A$, $\quad 21'$. $A \wedge B \longleftrightarrow B \wedge A$,
22. $(A \vee B) \vee C \longleftrightarrow A \vee (B \vee C)$, $\quad 22'$. $(A \wedge B) \wedge C \longleftrightarrow A \wedge (B \wedge C)$,
23. $A \vee (B \wedge C) \longleftrightarrow (A \vee B) \wedge (A \vee C)$,
$23'$. $A \wedge (B \vee C) \longleftrightarrow (A \wedge B) \vee (A \wedge C)$,
24. $A \vee A \longleftrightarrow A$, $\quad 24'$. $A \wedge A \longleftrightarrow A$,
25. $\sim (A \vee B) \longleftrightarrow \sim A \wedge \sim B$, $\quad 25'$. $\sim (A \wedge B) \longleftrightarrow \sim A \vee \sim B$,
26. $A \longrightarrow B \longleftrightarrow \sim A \vee B$,
27. $A \longrightarrow B \longleftrightarrow \sim (A \wedge \sim B)$,
28. $A \vee B \longleftrightarrow \sim A \longrightarrow B$,
29. $A \vee B \longleftrightarrow \sim (\sim A \wedge \sim B)$,
30. $A \wedge B \longleftrightarrow \sim (A \longrightarrow \sim B)$,
31. $A \wedge B \longleftrightarrow \sim (\sim A \vee \sim B)$,
32. $(A \longleftrightarrow B) \longleftrightarrow (A \longrightarrow B) \wedge (B \longrightarrow A)$.
33. $(A \longrightarrow B \vee C) \longleftrightarrow (A \wedge \sim B \longrightarrow C)$.

1.2 Логічне слідування в логіці висловлень

Логічний наслідок в логіці висловлень. Властивості логічного слідування. Доведення в логіці висловлень. Правила виведення. Методи доведення теорем.

1. Нехай $A_1, A_2, \ldots, A_n$ і B — деякі формули логіки висловлень.

Означення 1 *Висловлення B називається логічним наслідком висловлень $A_1, A_2, \ldots, A_n$ (це позначається як $A_1, A_2, \ldots, A_n \models B$), якщо для кожного розподілення істинностних значень, які приписуються логічним змінним, що входять хоч би в одну з формул $A_1, A_2, \ldots, A_n$ і B , формула B отримує значення* 1*, як тільки кожне A_i отримує значення* 1.

Надалі ми називатимемо формули $A_1, A_2, \ldots, A_n$ *посилками*, а формулу B — *наслідком*. Якщо виконується $A_1, A_2, \ldots, A_n \models B$, то ще кажуть, що B *логічно випливає* з посилок $A_1, A_2, \ldots, A_n$, і говорять, що дане міркування є коректним с точки зору логіки. Доведемо, наприклад, що $A, B, C \wedge A \longrightarrow\sim B \models\sim C$. Для цього складемо таблицю істинності:

A	B	C	A	B	$C \wedge A \longrightarrow\sim B$	$\sim C$
0	0	0	0	0	1	1
1	0	0	1	0	1	1
0	1	0	0	1	1	1
1	1	0	1	1	1	1
0	0	1	0	0	1	0
1	0	1	1	0	1	0
0	1	1	0	1	1	0
1	1	1	1	1	0	0

Ми бачимо, що в четвертому рядку всі посилки приймають значення 1 і значення наслідку також дорівнює 1. Згідно означення це означає, що наслідок $\sim C$ логічно випливає з посилок $A, B, C \wedge A \longrightarrow\sim B$.

Має місце така очевидна теорема:

Теорема 4. *Для довільних формул $A_1, A_2, \ldots, A_n$ і B має місце $A_1, A_2, \ldots, A_n \models B$ тоді і тільки тоді, коли $\models A_1 \wedge A_2 \wedge \ldots \wedge A_n \longrightarrow B$.*

Наслідок 1. *$A \models B$ тоді і тільки тоді, коли $\models A \longrightarrow B$ для довільних формул логіки висловлень.*

Наслідок 2. *Для довільних формул $A_1, A_2, \ldots, A_n$ і B має місце $A_1, A_2, \ldots, A_n \models B$ тоді і тільки тоді, коли $A_1, A_2, \ldots, A_{n-1} \models A_n \longrightarrow B$. В більш загальній формі $A_1, A_2, \ldots, A_n \models B$ тоді і тільки тоді, коли $\models A_1 \longrightarrow (A_2 \longrightarrow (\ldots (A_n \longrightarrow B) \ldots))$.*

2. Відмітимо дві очевидні властивості логічного слідування:

1. $A_1, A_2, \ldots, A_n \models A_i$ для кожного $i = 1, 2, \ldots, n$.
2. Якщо $A_1, A_2, \ldots, A_n \models B_j$ для $j = 1, 2, \ldots, p$ і якщо $B_1, B_2, \ldots, B_p \models C$, то $A_1, A_2, \ldots, A_n \models C$.

Доведення властивостей легко проводиться, якщо скористатись теоремою 4.

3. Нехай формула B є логічним наслідком з посилок $A_1, A_2, \ldots, A_n$, тоді можна побудувати так зване *доведення* (або інакше *виведення*) формули B з вказаних посилок, яке визначається означенням 2.

Означення 2 *Доведенням формули B з посилок $A_1, A_2, \ldots, A_n$ називається скінченна послідовність формул логіки висловлень, останньою з яких є формула B, причому наявність кожної формули E в цій послідовності обгрунтовується застосуванням одного з наступних правил:* [4]

- ***Правило посилки*** (*правило* $\mathbf{p}$)*: формула E є посилка.*
- ***Правило наслідку*** (*правило* $\mathbf{t}$)*: формулі E в послідовності передують такі формули $A, \ldots, C$, що $\models A \wedge \ldots \wedge C \longrightarrow E$.*
- ***Правило умовного відокремлення*** (*правило* $\mathbf{cp}$)*: формула $B \longrightarrow C$ виправдана в доведенні, посилками якого слугують формули $A_1, A_2, \ldots, A_n$, якщо встановлено, що C є логічний наслідок формул $A_1, A_2, \ldots, A_n, B$.*

Приклад 1. Довести, що $A \vee B, A \longrightarrow C, B \longrightarrow D \models C \vee D$, після чого побудувати доведення формули $C \vee D$ з посилок $A \vee B, A \longrightarrow C, B \longrightarrow D$.

Доведення. Згідно теореми 4 розглянемо формулу

$$(A \vee B) \wedge (A \longrightarrow C) \wedge (B \longrightarrow D) \longrightarrow (C \vee D) \tag{1.2.1}$$

і покажемо, що вона є тавтологія. Припустимо, що це не так, тобто

$$|(A \vee B) \wedge (A \longrightarrow C) \wedge (B \longrightarrow D) \longrightarrow (C \vee D)| = 0,$$

тому $|A \vee B| = 1$, $|A \longrightarrow C| = 1$, $|B \longrightarrow D| = 1$, $|C \vee D| = 0$. З останньої рівності маємо, що $|C| = 0$ і $|D| = 0$, тому $|A \longrightarrow 0| = 1$ і $|B \longrightarrow 0| = 1$. Таким чином, $|A| = |B| = 0$, тому $|A \vee B| = 0$, що протиріччить рівності $|A \vee B| = 1$. Отже, формула (1.2.1) є тавтологія, а це означає згідно теореми 4, що формула $C \vee D$ є логічним наслідком формул $A \vee B, A \longrightarrow C, B \longrightarrow D$.

Нижче дано доведення формули $C \vee D$ з посилок $A \vee B, A \longrightarrow C, B \longrightarrow D$.

1. $A \longrightarrow C$ (правило $\mathbf{p}$)
2. $A \vee B \longrightarrow C \vee B$ (правило $\mathbf{t}$, рядок 1, тавтологія 11)
3. $B \longrightarrow D$ (правило $\mathbf{p}$)
4. $C \vee B \longrightarrow C \vee D$ (правило $\mathbf{t}$, рядок 3, тавтологія 11)
5. $A \vee B \longrightarrow C \vee D$ (правило $\mathbf{t}$, рядки 2, 4, тавтологія 7)
6. $A \vee B$ (правило $\mathbf{p}$)

[4] Такі правила називають в математичній логіці *правилами виведення*.

7. $C \vee D$ (правило **t**, рядки 5, 6, тавтологія 1)

Приклад 1 розглянуто повністю. □

Багато теорем в математиці мають форму імплікації, причому припущеннями слугують аксіоми теорії, що розглядається. У символічній формі така теорема має наступний вид:

$$A_1, A_2, \ldots, A_n \models B \longrightarrow C,$$

де формули $A_1, A_2, \ldots, A_n$ — аксіоми, а $B \longrightarrow C$ — наслідок, який доводиться. Звичайний спосіб доведення такої теореми полягає в тому, що B приймають як ще одну посилку, яку називають *гіпотезою*, а потім виводять C як логічний наслідок. При цьому використовується той факт, що $A_1, A_2, \ldots, A_n \models B \longrightarrow C$ тоді і тільки тоді, коли $A_1, A_2, \ldots, A_n, B \models C$, що обгрунтовується наслідком 2 теореми 4, тобто користуються правилом умовного відокремлення.

Приклад 2. Довести, що $A \longrightarrow (B \longrightarrow C), \sim D \vee A, B \models D \longrightarrow C$.

1. $A \longrightarrow (B \longrightarrow C)$ (правило **p**)
2. $\sim D \vee A$ (правило **p**)
3. B (правило **p**)
4. D (гіпотеза, правило **p**)
5. A (правило **t**, рядки 2, 4, тавтологія 3)
6. $B \longrightarrow C$ (правило **t**, рядки 1, 5, тавтологія 1)
7. C (правило **t**, рядки 3, 6, тавтологія 1)
8. $D \longrightarrow C$ (правило **cp**, рядки 4, 7, див. властивість 1 на стор. 11)

Теорема 5. *Нехай $A_1, \ldots, A_n, B$ — формули логіки висловлень і T — довільна тавтологія, тоді $A_1, \ldots, A_n \models B$, якщо і тільки якщо $T, A_1, \ldots, A_n \models B$.*

Доведення. Справді, згідно теореми 4 твердження $A_1, \ldots, A_n \models B$ означає, що $\models A_1 \wedge \ldots \wedge A_n \longrightarrow B$. Оскільки $\models T$, то очевидно $|A_1 \wedge \ldots \wedge A_n| = |T \wedge A_1 \wedge \ldots \wedge A_n|$, а тому $\models T \wedge A_1 \wedge \ldots \wedge A_n \longrightarrow B$, що означає $T, A_1, \ldots, A_n \models B$. □

Теорема 5 дозволяє в доведення за правилом **p** вводити довільну тавтологію. На практиці крім вказаних вище трьох правил виведення користуються й іншими додатковими правилами виведення, що значно полегшує доведення багатьох теорем. Нижче наведені найбільш вживані правила виведення:

- **Модус поненс** (МП): $A, A \longrightarrow B \models B$ (див. тавтологію 1).
- **Модус толленс** (МТ): $\sim B, A \longrightarrow B \models \sim A$ (див. тавтологію 2).
- **Введення кон'юнкції** (ВК): $A, B \models A \wedge B$ (згідно тавт. 4 і наслідку 2 теореми 4).

- **Знищення кон'юнкції** (ЗК): $A \wedge B \models A$ або $A \wedge B \models B$ (див. тавтологію 5).
- **Введення диз'юнкції** (ВД): $A \models A \vee B$ або $B \models A \vee B$ (див. тавтологію 6).
- **Знищення диз'юнкції** (ЗД): $\sim A, A \vee B \models B$ або $\sim B, A \vee B \models A$ (див. тавт. 3).
- **Правило силогізму** (ПС): $A \longrightarrow B, B \longrightarrow C \models A \longrightarrow C$ (див. тавтологію 7).
- **Правило контрапозиції** (ПК): $A \longrightarrow B \models \sim B \longrightarrow \sim A$ (див. тавтологію 20).

Підсумовуючи сказане вище, ми можемо дати таке означення доведення:

Означення 3 *Доведенням формули B з посилок $A_1, A_2, \ldots, A_n$ називається скінченна послідовність формул логіки висловлень, останньою з яких є формула B, причому наявність кожної формули E в цій послідовності обгрунтовується застосуванням одного з наведених вище правил виведення до деяких формул, що передують їй у цій послідовності, або ж формула E є тавтологією.*

Приклад 3. Побудувати доведення теореми: $A \vee B \longrightarrow C \wedge D, D \vee E \longrightarrow F, A \models F$.

1. A (посилка) [5]
2. $A \vee B$ (ВД, 1)
3. $A \vee B \longrightarrow C \wedge D$ (посилка)
4. $C \wedge D$ (МП, 2, 3)
5. D (ЗК, 4)
6. $D \vee E$ (ВД, 5)
7. $D \vee E \longrightarrow F$ (посилка)
8. F (МП, 6, 7)

4. Розглянемо тепер деякі методи доведення теорем. Припустимо, що нам потрібно довести теорему:

$$A_1, A_2, \ldots, A_n \models B. \tag{1.2.2}$$

Якщо ми будуємо доведення цієї теореми, користуючись означеннями 2 або 3, тобто з посилок $A_1, A_2, \ldots, A_n$ виводимо формулу B, то таке доведення називається *прямим*. В розглянутих вище трьох прикладах нами були побудовані прямі доведення теорем. Крім прямого доведення часто на практиці користуються непрямими доведеннями, оскільки в багатьох випадках вони значно полегшують процес доведення теореми. Ми розглянемо зараз три методи непрямих доведень:

[5] Надалі ми будемо писати просто слово "посилка" замість слів "правило **p**".

Доведення від супротивного полягає в тому, що до посилок приєднується, так звана, гіпотеза $\sim B$. Після цього, користуючись означеннями 2 або 3 будується пряме доведення формули виду $H\wedge \sim H$, де H — деяка формула. Неважко бачити, що формула $H\wedge \sim H$ є протиріччям. Отже, насправді ми будуємо пряме доведення твердження

$$A_1, A_2, \ldots, A_n, \sim B \models H\wedge \sim H, \tag{1.2.3}$$

але стверджуємо, що нами доведене твердження (1.2.2). Останнє пояснюється таким чином. Твердження (1.2.3) згідно теореми 4 означає

$$\models (A_1 \wedge A_2 \wedge \ldots \wedge A_n)\wedge \sim B \longrightarrow H\wedge \sim H.$$

Оскільки $|H\wedge \sim H| = 0$, то очевидно $|(A_1 \wedge A_2 \wedge \ldots \wedge A_n)\wedge \sim B| = 0$ для всіх значень логічних змінних, які входять в дані формули. Отже,

$$\models\sim ((A_1 \wedge A_2 \wedge \ldots \wedge A_n)\wedge \sim B).$$

Таким чином, згідно тавтології 27 маємо твердження (1.2.2).

Доведення за правилом гіпотези застосовується тоді, коли наслідок, який випливає з посилок, має вид імплікації, тобто теорема має таку форму:

$$A_1, A_2, \ldots, A_n \models B \longrightarrow C. \tag{1.2.4}$$

В цьому випадку до посилок приєднується, так звана, гіпотеза B. Після цього, користуючись означеннями 2 або 3 будується пряме доведення формули C. Даний метод грунтується на застосування правила умовного відокремлення, тобто правила **ср**. Отже, щоб довести твердження 1.2.4, ми будуємо пряме доведення для твердження

$$A_1, A_2, \ldots, A_n, B \models C.$$

Доведення методом частинних випадків застосовується тоді, коли наслідок, який випливає з посилок, має вид диз'юнкції, тобто теорема має таку форму:

$$A_1, A_2, \ldots, A_n \models B \vee C. \tag{1.2.5}$$

В цьому випадку до посилок приєднується, так звана, гіпотеза $\sim B$ (або $\sim C$). Після цього, користуючись означеннями 2 або 3 будується пряме доведення формули C (або відповідно B), тобто доводиться твердження

$$A_1, A_2, \ldots, A_n, \sim B \models C \tag{1.2.6}$$

або відповідно $A_1, A_2, \ldots, A_n, \sim C \models B$. Обгрунтуємо тільки-що сказане. Дійсно, твердження (1.2.5) згідно теореми 4 означає

$$\models A_1 \wedge A_2 \wedge \ldots \wedge A_n \longrightarrow B \vee C.$$

Приймаючи до уваги тавтологію 33, ми отримуємо

$$\models A_1 \wedge A_2 \wedge \ldots \wedge A_n\wedge \sim B \longrightarrow C,$$

що означає (1.2.6).

Продемонструємо тепер вказані методи доведень на прикладах.

Приклад 4. Побудувати доведення теореми:

$$(A \longrightarrow B) \wedge (C \longrightarrow D), (B \longrightarrow E) \wedge (D \longrightarrow F), \sim (E \wedge F), A \longrightarrow C \models\sim A.$$

Доведення проведемо методом від супротивного.

1. $\sim\sim A$ (гіпотеза)
2. $\sim\sim A \longrightarrow A$ (тавтологія)
3. A (МП, 1, 2)
4. $A \longrightarrow C$ (посилка)
5. C (МП, 3, 4)
6. $(A \longrightarrow B) \wedge (C \longrightarrow D)$ (посилка)
7. $C \longrightarrow D$ (ЗК, 6)
8. D (МП, 5, 7)
9. $(B \longrightarrow E) \wedge (D \longrightarrow F)$ (посилка)
10. $D \longrightarrow F$ (ЗК, 9)
11. F (МП, 8, 10)
12. $\sim (E \wedge F)$ (посилка)
13. $\sim (E \wedge F) \longrightarrow\sim E \vee \sim F$ (тавтологія)
14. $\sim E \vee \sim F$ (МП, 12, 13)
15. $F \longrightarrow\sim\sim F$ (тавтологія)
16. $\sim\sim F$ (МП, 11, 15)
17. $\sim E$ (ЗД, 14, 16)
18. $B \longrightarrow E$ (ЗК, 9)
19. $\sim B$ (МТ, 17, 18)
20. $A \longrightarrow B$ (ЗК, 6)
21. $\sim A$ (МТ, 19, 20)
22. $A \wedge \sim A$ (ВК, 3, 21) $\square$

Приклад 5. Побудувати доведення теореми:

$$\sim A \vee B, C \longrightarrow \sim B \models A \longrightarrow \sim C.$$

Доведення будемо проводити за правилом гіпотези:

1. A (гіпотеза)
2. $\sim A \vee B$ (посилка)
3. $A \longrightarrow \sim\sim A$ (тавтологія)
4. $\sim\sim A$ (МП, 1, 3)
5. B (ЗД, 2, 4)
6. $B \longrightarrow \sim\sim B$ (тавтологія)
7. $\sim\sim B$ (МП, 5, 6)
8. $C \longrightarrow \sim B$ (посилка)
9. $\sim C$ (МТ, 7, 8) □

Приклад 6. Побудувати доведення теореми:

$$X \vee Y, X \longrightarrow Z, Y \longrightarrow W \models Z \vee W.$$

Доведення проводимо методом частинних випадків:

1. $\sim Z$ (гіпотеза)
2. $X \longrightarrow Z$ (посилка)
3. $\sim X$ (МТ, 1, 2)
4. $X \vee Y$ (посилка)
5. Y (ЗД, 3, 4)
6. $Y \longrightarrow W$ (посилка)
7. W (МП, 5, 6) □

1.3 Рівносильність формул логіки висловлень. Нормальні форми

Визначення рівносильності формул. Список основних рівносильностей. Елементарні кон'юнкції та диз'юнкції. Нормальні форми. Досконалі нормальні форми. Теореми про зображення булевих функцій за допомогою досконалих нормальних форм. Релейно-контактні схеми.

1. Нехай A і B є дві формули логіки висловлень. Будемо казати, що вони *рівносильні* і позначати цей факт $A \equiv B$, якщо і тільки якщо для довільного розподілу істинностних значень логічних змінних, що входять хоча б в одну з цих формул, дані формули приймають однакові істинностні значення. Наприклад, покажемо, що формула $\sim p \vee q$ рівносильна формулі $p \longrightarrow (q \vee s) \wedge (s \longrightarrow q)$.

p	q	s	$\sim p$	$\sim p \vee q$	$q \vee s$	$s \longrightarrow q$	$(q \vee s) \wedge (s \longrightarrow q)$	$p \longrightarrow (q \vee s) \wedge (s \longrightarrow q)$
0	0	0	1	1	0	1	0	1
1	0	0	0	0	0	1	0	0
0	1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1
0	0	1	1	1	1	0	0	1
1	0	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1

В таблиці ми бачимо, що стовпці, які відповідають даним формулам однакові, що означає Рівносильність формул. Неважко також бачити, що відношення рівносильності між формулами є відношення еквівалентності, тобто воно рефлексивне, симетричне і транзитивне.

Теорема 6. *Для довільних формул A і B логіки висловлень $A \equiv B$ тоді і тільки тоді, коли $\models A \longleftrightarrow B$.*

Доведення. Дійсно, для довільного розподілу істинностних значень логічних змінних, що входять хоча б в одну з формул A, B (тобто які входять в формулу $A \longleftrightarrow B$), формули A і B приймають однакові значення, тобто $|A| = |B|$, що означає згідно означення еквівалентності, що $|A \longleftrightarrow B| = 1$. Таким чином, $\models A \longleftrightarrow B$. □

Теорема 7. *Якщо формули A, B, C, D такі, що $A \equiv B$ і D отримується з формули C підстановкою B замість одного або більшого числа входжень A, то $C \equiv D$.*

Доведення теореми очевидне.

Теорема 8. *В логіці висловлень мають місце такі рівносильності:*

1. $\overline{\overline{A}} \equiv A$ — закон подвійного заперечення, [6]

2. $(AB)C \equiv A(BC)$ — асоціативність кон'юнкції, [7]

[6] Нагадаємо, що $\overline{A}$ означає $\sim A$.

[7] AB означає кон'юнкцію $A \wedge B$.

3. $(A \vee B) \vee C \equiv A \vee (B \vee C)$ — *асоціативність диз'юнкції,*

4. $AB \equiv BA$ — *комутативність кон'юнкції,*

5. $A \vee B \equiv B \vee A$ — *комутативність диз'юнкції,*

6. $AA \equiv A$ — *ідемпотентність кон'юнкції,*

7. $A \vee A \equiv A$ — *ідемпотентність диз'юнкції,*

8. $A(B \vee C) \equiv AB \vee AC$ — *перший закон дистрибутивності,*

9. $A \vee BC \equiv (A \vee B)(A \vee C)$ — *другий закон дистрибутивності,*

10. $\overline{AB} \equiv \overline{A} \vee \overline{B}$ — *перший закон де Моргана,*

11. $\overline{A \vee B} \equiv \overline{A}\,\overline{B}$ — *другий закон де Моргана,*

12. $A \longrightarrow B \equiv \overline{A} \vee B$,

13. $A \longleftrightarrow B \equiv (A \longrightarrow B)(B \longrightarrow A)$,

14. $A \vee \overline{A} \equiv 1$ — *закон виключеного третього,*

15. $A\,\overline{A} \equiv 0$ — *закон протиріччя,*

16. $A\,1 \equiv A$, $A \vee 1 \equiv 1$, $A\,0 \equiv 0$, $A \vee 0 \equiv A$ — *дії з константами,*

17. $A \longrightarrow B \equiv \overline{B} \longrightarrow \overline{A}$ — *закон контрапозиції.*

Рівносильності 1–17 доводяться за допомогою таблиць істинності і використовуються для спрощення формул логіки висловлень.

Приклад 1. Спростити формулу $(A \longrightarrow B)(A \vee BC)(A \longrightarrow C) \vee \overline{C}$.

Маємо

$$
\begin{aligned}
& (A \longrightarrow B)(A \vee BC)(A \longrightarrow C) \vee \overline{C} \overset{12}{\equiv} (\overline{A} \vee B)(A \vee BC)(A \longrightarrow C) \vee \overline{C} \overset{12}{\equiv} \\
\equiv\ & (\overline{A} \vee B)(A \vee BC)(\overline{A} \vee C) \vee \overline{C} \overset{4}{\equiv} (\overline{A} \vee B)(\overline{A} \vee C)(A \vee BC) \vee \overline{C} \overset{9}{\equiv} \\
\equiv\ & (\overline{A} \vee BC)(A \vee BC) \vee \overline{C} \overset{9}{\equiv} (\overline{A}A \vee BC) \vee \overline{C} \overset{15}{\equiv} (0 \vee BC) \vee \overline{C} \overset{16}{\equiv} \\
\equiv\ & BC \vee \overline{C} \overset{9}{\equiv} (B \vee \overline{C})(C \vee \overline{C}) \overset{14}{\equiv} (B \vee \overline{C})\,1 \overset{16}{\equiv} B \vee \overline{C} \overset{5}{\equiv} \overline{C} \vee B \overset{12}{\equiv} C \longrightarrow B. \qquad \square
\end{aligned}
$$

2. *Елементарною кон'юнкцією* називається кожна кон'юнкція скінченного числа попарно різних логічних змінних, взятих із запереченням або без нього. Наприклад, xyz, $x\bar{y}z\bar{u}$, $x_1\bar{x}_2\bar{x}_3\bar{x}_4x_5$ є елементарні кон'юнкції, а $x\bar{x}y$, $xyz\bar{y}$ такими не будуть. *Елементарною диз'юнкцією* називається диз'юнкція скінченного числа попарно різних логічних змінних, взятих із запереченням або без нього. Наприклад, $x \vee y \vee z$, $\bar{x} \vee y \vee \bar{z}$, $x_1 \vee \bar{x}_2 \vee x_3 \vee \bar{x}_4$, а $x \vee y \vee \bar{x}$ вже не є елементарною диз'юнкцією.

Зафіксуємо деяку множину X логічних змінних. Елементарні кон'юнкції, які містять всі змінні з X, будемо називати *конституентами одиниці над* X, відповідно елементарні диз'юнкції, які задовольняють подібну властивість, назвемо *конституентами нуля над* X. Наприклад, якщо $X = \{x, y, z, u\}$, то $x\bar{y}z\bar{u}$, $xyz\bar{u}$ — конституенти одиниці над X, $\bar{x} \vee y \vee z \vee \bar{u}$ — конституента нуля над X. Неважко бачити, що кожна конституента одиниці (відповідно, конституента нуля) тільки на одному, єдиному для неї, двійковому наборі приймає значення 1 (відповідно, значення 0). В цьому випадку говорять, що конституента відповідає даному двійковому набору. Наприклад, якщо $X = \{x, y, z\}$, то $x\bar{y}\bar{z}$ відповідає двійковому набору $(1, 0, 0)$, $\bar{x}yz$ — набору $(0, 1, 1)$, $x \vee y \vee \bar{z}$ відповідає $(0, 0, 1)$, а $\bar{x} \vee y \vee z$ — $(1, 0, 0)$.

Означення 4 *Диз'юнктивною (кон'юнктивною) нормальною формою називається диз'юнкція (кон'юнкція) скінченного числа попарно різних елементарних кон'юнкцій (диз'юнкцій).*

Наприклад, $xy \vee \bar{x} \vee y\bar{z} \vee yzu$, $xy\bar{z} \vee u$ є диз'юнктивні нормальні форми (скорочено ДНФ), а $(x \vee y)(\bar{x} \vee \bar{u} \vee z)$, $(x \vee \bar{y} \vee z)x$ — кон'юнктивні нормальні форми (скорочено КНФ). Надалі елементарні кон'юнкції в ДНФ будемо називати *доданками*, а елементарні диз'юнкціїї в КНФ — *множниками*.

ДНФ (КНФ) називається *досконалою*, якщо всі її доданки (множники) є конституенти одиниці над множиною всіх її логічних змінних. Досконалу ДНФ (КНФ) ми будемо скорочено позначати як ДДНФ (ДКНФ). Наприклад, $xy\bar{z} \vee \bar{x}y\bar{z}$ є ДДНФ, а $(x \vee \bar{y} \vee z)(x \vee y \vee \bar{z})(\bar{z} \vee \bar{y} \vee \bar{z})$ — ДКНФ. Відмітимо, що за допомогою очевидної рівносильності

$$Ax \vee A\bar{x} \equiv A, \tag{1.3.1}$$

де A — довільна формула, а x — логічна змінна, кожна ДНФ може бути зведена до ДДНФ, а з допомогою рівносильності

$$(A \vee x)(A \vee \bar{x}) \equiv A \tag{1.3.2}$$

кожна КНФ може бути зведена до ДКНФ.

Приклад 2. Звести формулу $xy \vee x\bar{z}$ до ДДНФ.

Маємо, згідно формули (1.3.1): $xy \vee x\bar{z} \equiv xyz \vee xy\bar{z} \vee x\bar{z} \equiv xyz \vee xy\bar{z} \vee xy\bar{z} \vee x\bar{y}\bar{z} \equiv xyz \vee xy\bar{z} \vee x\bar{y}\bar{z}$. □

Приклад 3. Звести формулу $(x \vee y)y$ до ДКНФ.

Згідно формули (1.3.2) маємо: $(x \vee y)y \equiv (x \vee y)(x \vee y)(\bar{x} \vee y) \equiv (x \vee y)(\bar{x} \vee y)$. □

На завершення відмітимо, що ДДНФ тавтології, що буде доведено пізніше, містить точно 2^n доданків, де n — число всіх логічних змінних даної тавтології. Це дає нам ще один спосіб перевірки формули на тавтологічність. Наприклад, доведемо, що формула $\underline{x(x \longrightarrow y)} \longrightarrow y$ є тавтологія. Отже, маємо: $x(x \longrightarrow y) \longrightarrow y \equiv x(\bar{x} \vee y) \longrightarrow y \equiv \overline{x(\bar{x} \vee y)} \vee y \equiv \bar{x} \vee \overline{\bar{x} \vee y} \vee y \equiv \bar{x} \vee x\bar{y} \vee y \equiv \bar{x}y \vee \bar{x}\bar{y} \vee x\bar{y} \vee y \equiv \bar{x}t \vee \bar{x}\bar{y} \vee x\bar{y} \vee xy \vee \bar{x}y \equiv$

$\bar{x}y \vee \bar{x}\bar{y} \vee x\bar{y} \vee xy$, звідки видно, що ДДНФ даної формули має чотири доданки, тобто вона є тавтологія.

Аналогічно, ДКНФ протиріччя, тобто тотожно хибної формули, містить точно 2^n множників, де n — число всіх змінних даної формули.

3. Виникає природне питання про те, чи можна кожну булеву функцію зобразити формулою алгебри висловлень. Позитивна відповідь на це питання випливає з наступних двох теорем.

Теорема 9. *Кожну булеву функцію $f(x_1, x_2, \ldots, x_n)$ можна зобразити наступною формулою логіки висловлень:*

$$f(x_1, x_2, \ldots, x_n) = \bigvee_{\vec{a}=(0,\ldots,0)}^{(1,\ldots,1)} f(a_1, a_2, \ldots, a_n) x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}, \tag{1.3.3}$$

де $\vec{a} = (a_1, a_2, \ldots, a_n)$, $a_i \in \{0, 1\}$, $x_i^0 = \bar{x}_i$, $x_i^1 = x_i$ для кожного $i = 1, 2, \ldots, n$.

Доведення. Покажемо, що ліва і права частини співвідношення (1.3.3) співпадають. Нехай $\vec{a} = (a_1, a_2, \ldots, a_n)$ є довільний двійковий набір. Підставляючи його в (1.3.3) в лівій частині будемо мати $f(a_1, a_2, \ldots, a_n)$, а в правій отримаємо:

$$\bigvee_{\vec{a}=(0,\ldots,0)}^{(1,\ldots,1)} f(a_1, a_2, \ldots, a_n) a_1^{a_1} a_2^{a_2} \ldots a_n^{a_n} = f(a_1, a_2, \ldots, a_n) a_1^{a_1} a_2^{a_2} \ldots a_n^{a_n} = f(a_1, a_2, \ldots, a_n),$$

оскільки $x_i^{a_i} = 1$ лише при $x_i = a_i$. □

Наслідок 3. *Кожна булева функція, яка тотожно не дорівнює нулеві, може бути єдиним чином зображена в ДДНФ.*

Теорема 10. *Кожну булеву функцію $f(x_1, x_2, \ldots, x_n)$ можна зобразити наступною формулою логіки висловлень:*

$$f(x_1, x_2, \ldots, x_n) = \bigwedge_{\vec{a}=(0,\ldots,0)}^{(1,\ldots,1)} \left(f(a_1, a_2, \ldots, a_n) \vee x_1^{\bar{a}_1} \vee x_2^{\bar{a}_2} \vee \ldots \vee x_n^{\bar{a}_n} \right), \tag{1.3.4}$$

де $\vec{a} = (a_1, a_2, \ldots, a_n)$, $a_i \in \{0, 1\}$, $x_i^0 = \bar{x}_i$, $x_i^1 = x_i$ для кожного $i = 1, 2, \ldots, n$.

Доведення. За законом подвійного заперечення маємо

$$f(x_1, x_2, \ldots, x_n) = \overline{\overline{f(x_1, x_2, \ldots, x_n)}}, \tag{1.3.5}$$

а за теоремою 9 можемо записати рівність

$$\overline{f(x_1, x_2, \ldots, x_n)} = \bigvee_{\vec{a}=(0,\ldots,0)}^{(1,\ldots,1)} \overline{f(a_1, a_2, \ldots, a_n)} x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n}. \tag{1.3.6}$$

Підставляючи (1.3.6) в (1.3.5) і застосовуючи закон де Моргана ми отримаємо:

$$f(x_1,x_2,\dots,x_n)=\overline{\bigvee_{\vec{a}=(0,\dots,0)}^{(1,\dots,1)}\overline{f(a_1,a_2,\dots,a_n)}x_1^{a_1}x_2^{a_2}\dots x_n^{a_n}}=$$

$$=\bigwedge_{\vec{a}=(0,\dots,0)}^{(1,\dots,1)}\left(f(a_1,a_2,\dots,a_n)\vee\overline{x_1^{\bar{a}_1}}\vee\overline{x_2^{\bar{a}_2}}\vee\dots\vee\overline{x_n^{\bar{a}_n}}\right)=$$

$$=\bigwedge_{\vec{a}=(0,\dots,0)}^{(1,\dots,1)}\left(f(a_1,a_2,\dots,a_n)\vee x_1^{\bar{a}_1}\vee x_2^{\bar{a}_2}\vee\dots\vee x_n^{\bar{a}_n}\right),$$

оскільки $\overline{x_1^{a_i}}=x_i^{\bar{a}_i}$. □

Наслідок 4. *Кожна булева функція, яка тотожно не дорівнює одиниці, може бути єдиним чином зображена в ДКНФ.*

Приклад 4. Знайти ДДНФ і ДКНФ для булевої функції $f(x,y,z)$, яка задана такою таблицею істинності:

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

Спочатку знайдемо ДДНФ для даної функції. Згідно теореми 9 відмітимо рядки, в яких функція приймає значення 1. Кожній такій одиниці відповідає певний двійковий набір. Це будуть такі набори: $(1,0,0)$, $(0,1,0)$, $(1,0,1)$. Кожному такому двійковому набору, як відомо, відповідає конституента одиниці, а саме: $x\bar{y}\bar{z}$, $\bar{x}y\bar{z}$, $x\bar{y}z$. Отже, ДДНФ для даної функції є диз'юнкція цих конституент одиниці, тобто

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

Щоб знайти ДКНФ для вказаної булевої функції, треба відмітити рядки, в яких значення функції дорівнює 0. Таких рядків п'ять. Далі необхідно відмітити в цих рядках двійкові набори: $(0,0,0)$, $(1,1,0)$, $(0,0,1)$, $(0,1,1)$, $(1,1,1)$. Після цього для кожного такого двійкового набору записати відповідну конституенту нуля: $x\vee y\vee z$, $\bar{x}\vee\bar{y}\vee z$, $x\vee y\vee\bar{z}$, $x\vee\bar{y}\vee\bar{z}$, $\bar{x}\vee\bar{y}\vee\bar{z}$. За теоремою 10 дана функція є кон'юнкція останніх конституент нуля, тобто

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

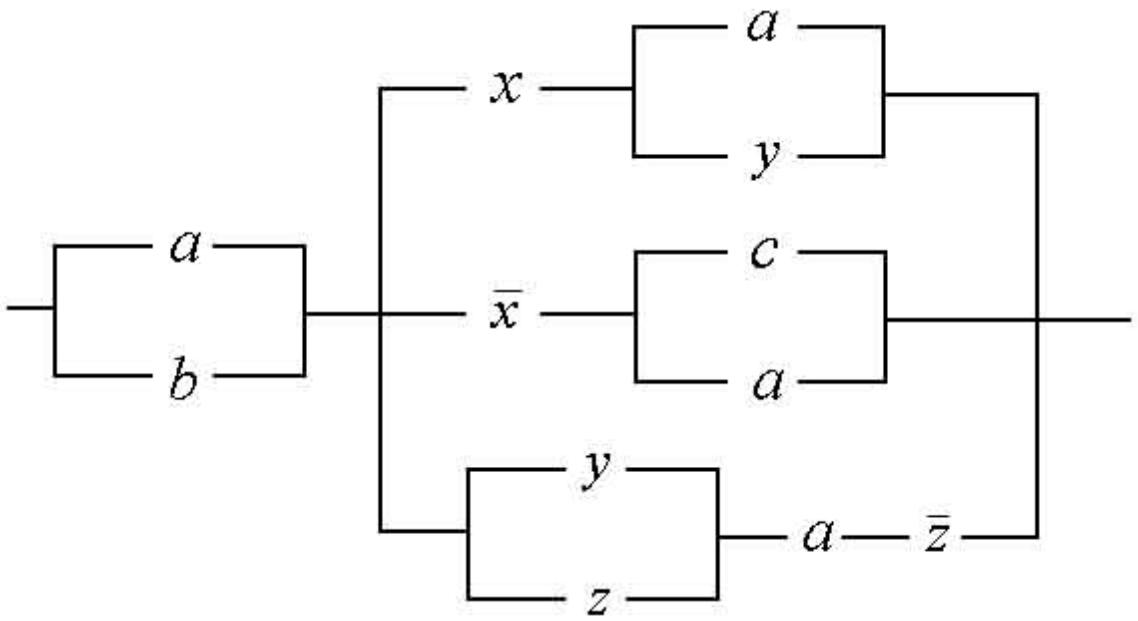
4. Під *релейно-контактною схемою* розуміють пристрій, який складається з провідників і двопозиційних контактів. *Двопозиційний контакт* — це фізичне тіло, яке може перебувати лише в двох станах — "ввімкнено" або "вимкнуто", які будемо позначати через 1 і 0 відповідно. Контакти ми позначатимемо малими латинськими літерами.

Між собою контакти можуть з'єднуватись послідовно і паралельно. При послідовному з'єднанні контактів сигнал через з'єднання проходить тоді і тільки тоді, коли він проходить через кожний контакт. При паралельному ж з'єднанні контактів сигнал

проходить тоді і тільки тоді, коли ввімкнено хоча б один із контактів. Таким чином, послідовне з'єднання контактів відповідає кон'юнкції двох логічних змінних, а паралельне — диз'юнкції. Отже, послідовне з'єднання контактів a і b ми позначатимемо через ab, а паралельне — через $a \vee b$. Далі, через $\bar{a}$ будемо позначати такий контакт, який проводить сигнал тоді і тільки тоді, коли контакт a його не проводить. Таким чином, будь-яка булева функція може бути реалізована за допомогою релейно-контактної схеми, оскільки її можна зобразити формулою алгебри висловлень, наприклад, ДДНФ або ДКНФ, в яких використовуються лише три логічних операції — кон'юнкція, диз'юнкція і заперечення. Так булева функція, яка визначається формулою логіки висловлень

$$(a \vee b)\Big(x(a \vee y) \vee \bar{x}(c \vee a) \vee (y \vee z)a\bar{z}\Big),$$

зображується наступною релейно-контактною схемою:



1.4 Повні системи булевих функцій. Алгебра Жегалкіна

Повні системи булевих функцій. Теорема про число повних бінарних логічних операцій. Алгебра Жегалкіна. Поліном Жегалкіна. Теорема про зображення булевої функції поліномом Жегалкіна

1. Система булевих функцій $\{f_1, f_2, \ldots, f_n\}$ називається *повною*, якщо довільна булева функція може бути подана за допомогою перейменування аргументів і суперпозиції функцій $f_1, f_2, \ldots, f_n$, взятих довільне скінченне число разів. Наприклад, система функцій $\{\sim, \wedge, \vee\}$ є повною, оскільки, як відомо з попередньої лекції, кожна булева функція може бути подана ДДНФ або ДКНФ. Популярно означення повноти можна дати таким чином: *множина логічних операцій* Φ *називається повною, якщо кожна булева функція може бути задана формулою, яка записана за допомогою лише операцій з* Φ. Як приклад неповної системи можна навести систему, яка складається з однієї операції заперечення, тобто $\sim$. Це пояснюється тим, що за допомогою одного заперечення можна побудувати лише дві функції — логічну змінну та її заперечення. Чи існують крім $\{\sim, \wedge, \vee\}$ інші повні системи булевих функцій? Відповідь на це дається у наступній теоремі.

Теорема 11. *Системи логічних операцій* $\{\sim, \wedge\}$, $\{\sim, \vee\}$, $\{\,|\,\}$, $\{\downarrow\}$ *(див. стор. 8) є повними.*

Доведення. Оскільки $\{\sim, \wedge, \vee\}$ є повна система логічних операцій і мають місце рівності:

$$\begin{aligned} &x \vee y = \overline{\bar{x} \wedge \bar{y}}, && x \wedge y = \overline{\bar{x} \vee \bar{y}}, \\ &\bar{x} = x \,|\, x, && x \vee y = (x \,|\, x) | (y \,|\, y), \\ &\bar{x} = x \downarrow x, && x \wedge y = (x \downarrow x) \downarrow (y \downarrow y), \end{aligned}$$

то очевидно наведені системи операцій будуть повними.[8] □

2. З теореми 11 випливає, що за допомогою лише однієї операції штриха Шеффера або лише однієї стрілки Пірса можна зобразити формулою довільну булеву функцію. Виникає питання про число булевих функцій, які мають подібну властивість. Такі булеві функції ми надалі будемо називати *повними*. У випадку бінарних логічних операцій має місце така теорема:

Теорема 12. *Єдиними бінарними повними логічними операціями є штрих Шеффера і стрілка Пірса.*

Доведення. Припустимо, що бінарна логічна операція $h(x, y)$ є повною. Як би $h(1, 1) = 1$, то довільна булева функція, яку можна зобразити за допомогою лише операції $h(x, y)$, приймала би значення 1, коли всі її аргументи приймали б значення 1. Але ж тоді функція $\bar{x}$ не могла б бути виражена через $h(x, y)$. Отже, $h(1, 1) = 0$. Аналогічними міркуваннями ми доводимо, що $h(0, 0) = 1$. Отже, ми маємо:

[8] Доведіть, що мають місце рівності: $x \wedge y = (x \,|\, y) | (x \,|\, y)$, $x \vee y = (x \downarrow y) \downarrow (x \downarrow y)$.

x	y	$h(x,y)$
1	1	0
0	1	
1	0	
0	0	1

Залишаються незаповненими другий і третій рядки. Для них можливі такі випадки:

$$\text{а)}\ \begin{matrix}0\\0\end{matrix};\quad \text{б)}\ \begin{matrix}1\\1\end{matrix};\quad \text{в)}\ \begin{matrix}0\\1\end{matrix};\quad \text{г)}\ \begin{matrix}0\\0\end{matrix}.$$

Випадок а) визначає стрілку Пірса, б) — штрих Шеффера, в) — заперечення y, тобто $\bar{y}$, г) — заперечення x, тобто $\bar{x}$. Однак, заперечення не є повною булевою операцією, тому залишаються лише пункти а) і б). □

3. Нехай F є множина всіх булевих функцій. На множині F можна також розглядати логічні операції, які для булевих функцій вводяться таким чином. Нехай $f(x_1,\dots,x_n)$ і $g(y_1,\dots,y_m)$ є деякі булеві функції, $\top$ — деяка логічна операція. Тоді через $f\top g$ будемо позначати булеву функцію, множиною аргументів якої є об'єднання аргументів даних функцій, а значення функції для цих аргументів визначається згідно рівності:

$$(f\top g)(x_1,\dots,y_m)\stackrel{df}{=} f(x_1,\dots,x_n)\top g(y_1,\dots,y_m).$$

Наприклад, кон'юнкція функцій $f(x,y,z)$ і $g(y,u,z,v)$ є функція виду $(f\wedge g)(x,y,z,u,v)$, значення якої визначаються згідно рівності:

$$(f\wedge g)(x,y,z,u,v)=f(x,y,z)\wedge g(y,u,z,v).$$

Аналогічним чином можна визначити довільні логічні операції для булевих функцій.

Алгеброю Жегалкіна ми будемо називати алгебру виду $(\Phi,\wedge,\oplus)$, де $\Phi=F\cup\{0,1\}$ — множина всіх булевих функцій, доповнена константами 0 і 1, $\wedge$ — операція кон'юнкції, $\oplus$ — сума по модулю 2. За допомогою таблиць істинності легко доводиться така теорема:

Теорема 13. *В алгебрі Жегалкіна виконуються такі співвідношення:*

1. $(xy)z=x(yz)$ *—асоціативність кон'юнкції;*
2. $(x\oplus y)\oplus z=x\oplus(y\oplus z)$ *— асоціативність суми по модулю 2;*
3. $xy=yx$ *— комутативність кон'юнкції;*
4. $x\oplus y=y\oplus x$ *— комутативність суми по модулю 2;*
5. $xx=x$ *— ідемпотентність кон'юнкції;*
6. $x\oplus x=0$ *— закон зведення подібних членів;*
7. $x(y\oplus z)=xy\oplus xz$ *— закон дистрибутивності;*
8. $x\cdot 1=x,\ x\cdot 0=0,\ x\oplus 0=x$ *— дії з константами.*

Нехай задана алгебра Жегалкіна. Кон'юнкція довільного скінченного числа попарно різних логічних змінних називається *одночленом*. Наприклад, 1, 0, x, y, xy, xyz, xu. Далі, сума по модулю 2 скінченного числа попарно різних одночленів називається

поліномом Жегалкіна. Наприклад, $xz \oplus xy \oplus x \oplus z \oplus 1$, $xyzu \oplus xy \oplus z$. Сума по модулю 2 скінченного числа попарно різних конституент одиниці над деякою множиною логічних змінних називається *досконалою бісумарною нормальною формою* (скорочено ДБНФ). Наприклад, $xy\bar{z} \oplus x\bar{y}\bar{z} \oplus xyz \oplus \bar{x}\bar{y}\bar{z}$ — ДБНФ. Враховуючи, що $x \oplus 0 = x \vee 0$, виконується така теорема:

Теорема 14. *Кожна булева функція, яка тотожно не дорівнює нулеві, може бути єдиним чином зображена ДБНФ.*

Доведення проводиться так, як і для ДДНФ, тому ми його не наводимо. Відмітимо лише, що враховуючи той факт, що конституента одиниці приймає значення 1 тільки на єдинному двійковому наборі, який їй відповідає, а також, що $1 \oplus 0 = 1 \vee 0$, робимо висновок, що для отримання ДБНФ достатньо в ДДНФ замінити всі символи операції $\vee$ на символ $\oplus$. Наприклад, якщо $f(x, y, z) = xy\bar{z} \vee x\bar{y}\bar{z} \vee \bar{x}y\bar{z}$, то $f(x, y, z) = xy\bar{z} \oplus x\bar{y}\bar{z} \oplus \bar{x}y\bar{z}$.

4. ДБНФ дає нам можливість записувати формули у вигляді полінома Жегалкіна. Наприклад, нехай потрібно записати диз'юнкцію $x \vee y$ поліномом Жегалкіна. Запишемо спочатку дану диз'юнкцію в ДДНФ, тобто $x \vee y = xy \vee x\bar{y} \vee \bar{x}y$. Тоді ДБНФ має вигляд: $x \vee y = xy \oplus x\bar{y} \oplus \bar{x}y$. Оскільки $\bar{x} = x \oplus 1$, то

$$x \vee y = xy \oplus x(y \oplus 1) \oplus (x \oplus 1)y = xy \oplus xy \oplus x \oplus xy \oplus y = xy \oplus x \oplus y.$$

Отже, $xy \oplus x \oplus y$ — поліном Жегалкіна для $x \vee y$. Виникає питання: "Скільки має булева функція різних поліномів Жегалкіна, що її зображують?"Відповідь дається у наступній теоремі:

Теорема 15. *Кожна булева функція може бути єдиним чином зображена поліномом Жегалкіна.*

Доведення. Нехай $f(x_1, \ldots, x_n)$ є довільна булева функція. Оскільки константи 0 і 1 самі по собі є поліномами Жегалкіна, то ми можемо обмежитись випадком, коли $f(x_1, \ldots, x_n) \not\equiv 0$. Зобразимо цю функцію в ДБНФ, далі в цій формі замінимо вирази типу $\bar{x}$ на $x \oplus 1$, після чого, користуючись законами теореми 14, зводимо формулу до полінома Жегалкіна.

Доведемо його єдинність. Нехай існує два поліноми φ і ψ, які зображують одну і ту ж функцію $f(x_1, \ldots, x_n)$. Оскільки φ і ψ різні поліноми, то в одному з них знайдеться такий одночлен, якого немає в другому. Виберемо з таких одночленів той, який має найменшу кількість змінних. Таким чином, всі одночлени з меншим числом змінних, присутні в обох поліномах. Нехай вибраний нами одночлен позначається літерою P. Задамо тепер двійковий набір таким чином, щоб всі змінні, які входять в P, приймали значення 1, а всі інші змінні — 0. Тоді кожний одночлен, відмінний від P і який має більше ніж в P число змінних, приймає значення 0. Сам же одночлен P приймає значення 1. Таким чином, один з поліномів приймає значення $\varphi_0 \oplus 1$, а інший — ψ_0. Але $\varphi_0 = \psi_0$, оскільки це є значення однакової частини обох поліномів. Тому $\varphi_0 \oplus 1 \neq \psi_0$, що протиріччить тому, що поліноми зображують одну і ту ж функцію. Теорема доведена. □

З даної теореми випливає ще один спосіб доведення рівносильності формул. А саме, якщо дві формули зводяться до одного і того ж полінома Жегалкіна, то вони рівносильні. Доведемо, наприклад, що

$$(A \longrightarrow B) \longrightarrow (A \longrightarrow C) \equiv A \longrightarrow (B \longrightarrow C).$$

Маємо $x \longrightarrow y \equiv \bar{x} \vee y \equiv \bar{x}y \vee \bar{x}\bar{y} \vee xy \vee \bar{x}y \equiv \bar{x}y \vee \bar{x}\bar{y} \vee xy \equiv (x \oplus 1)y \oplus (x \oplus 1)(y \oplus 1) \oplus xy \equiv xy \oplus y \oplus xy \oplus x \oplus y \oplus 1 \oplus xy \equiv xy \oplus x \oplus 1$. Отже, $x \longrightarrow y \equiv xy \oplus x \oplus 1$. Таким чином,

$$\begin{aligned}
&(A \longrightarrow B) \longrightarrow (A \longrightarrow C) \equiv (AB \oplus A \oplus 1) \longrightarrow (AC \oplus A \oplus 1) \equiv \\
&\equiv (AB \oplus A \oplus 1)(AC \oplus A \oplus 1) \oplus (AB \oplus A \oplus 1) \oplus 1 \\
&\equiv ABC \oplus AB \oplus AB \oplus AC \oplus A \oplus A \oplus AC \oplus A \oplus 1 \oplus AB \oplus A \oplus 1 \oplus 1 \equiv \\
&\equiv ABC \oplus AB \oplus 1;
\end{aligned}$$

$$\begin{aligned}
&A \longrightarrow (B \longrightarrow C) \equiv A \longrightarrow (BC \oplus B \oplus 1) \equiv \\
&\equiv A(BC \oplus B \oplus 1) \oplus A \oplus 1 \equiv ABC \oplus AB \oplus A \oplus A \oplus 1 \equiv \\
&\equiv ABC \oplus AB \oplus 1.
\end{aligned}$$

Отже, обидві формули зводяться до одного і того ж полінома Жегалкіна, а це й означає, що вони рівносильні.

1.5 Замкнені класи булевих функцій. Теорема про функціональну повноту

Функції, які зберігають константи. Самодвоїсті функції. Монотонні функції. Лінійні функції. Теорема Поста про функціональну повноту системи булевих функцій.

1. Будемо казати, що булева функція $f(x_1, \ldots, x_n)$ *зберігає константу нуль* (відповідно, *одиницю*), якщо $f(0, \ldots, 0) = 0$ (відповідно, $f(1, \ldots, 1) = 1$). Множину всіх функцій, які зберігають ноль, позначимо через K_0, а які зберігають одиницю — через K_1.

Лема 1. *Суперпозиція функцій, які зберігають константу ноль, є знову функція, яка зберігає константу ноль. Інакше кажучи, клас функцій K_0 замкнений відносно суперпозицій.*

Доведення. Справді, нехай $f, f_1, \ldots, f_n \in K_0$. Розглянемо функцію $F(x_1, \ldots, x_n)$, яка є суперпозицією даних функцій і визначається рівністю:

$$F(x_1, \ldots, x_n) = f(f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n)). \tag{1.5.1}$$

Покажемо, що ця функція належить класу K_0. Маємо

$$F(0, \ldots, 0) = f(f_1(0, \ldots, 0), \ldots, f_n(0, \ldots, 0)) = f(0, \ldots, 0) = 0.$$

Отже, $F(x_1, \ldots, x_n) \in K_0$. □

Наслідок 5. *Повна система булевих функцій містить хоча б одну функцію, яка не зберігає константу ноль.*

Справді, якщо припустити, що всі функції повної системи зберігають константу ноль, то звідси випливатиме, що будь-яка булева функція також зберігає константу ноль. Але ж це не так, оскільки, наприклад, імплікація ноль не зберігає $(0 \longrightarrow 0 = 1)$, тобто $\longrightarrow \notin K_0$.

Аналогічно доводяться такі твердження:

Лема 2. *Суперпозиція функцій, які зберігають константу одиниця, є знову функція, яка зберігає константу одиниця. Інакше кажучи, клас функцій K_1 замкнений відносно суперпозицій.*

Наслідок 6. *Повна система булевих функцій містить хоча б одну функцію, яка не зберігає константу одиниця.*

Всі міркування аналогічні. Зауважимо лише, що сума по молулю два не зберігає одиницю $(1 \oplus 1 = 0)$, тобто $\oplus \notin K_1$.

2. Булева функція $f(x_1, \ldots, x_n)$ називається *самодвоїстою*, якщо вона на довільній парі протилежних двійкових наборів [9] приймає протилежні значення, тобто вона задовольняє рівність:

$$\overline{f(x_1, \ldots, x_n)} = f(\bar{x}_1, \ldots, \bar{x}_n) \tag{1.5.2}$$

[9] Двійкові набори виду $(x_1, \ldots, x_n)$ і $(\bar{x}_1, \ldots, \bar{x}_n)$ називаються *протилежними*, наприклад, $(0, 1, 1, 0, 1, 0)$ і $(1, 0, 0, 1, 0, 1)$ — приклад пари протилежних двійкових наборів.

для довільних $x_1, \ldots, x_n \in \{0, 1\}$.

Множину всіх самодвоїстих булевих функцій позначимо через K_s.

Лема 3. *Суперпозиція самодвоїстих булевих функцій є знову самодвоїста функція, тобто клас K_s замкнений відносно суперпозицій.*

Доведення. Дійсно, нехай $f, f_1, \ldots, f_m \in K_s$. Розглянемо функцію $F(x_1, \ldots, x_n)$, яка визначається рівністю (1.5.1). Тоді ми будемо мати:

$$\begin{aligned}\overline{F(x_1, \ldots, x_n)} &= \overline{f(f_1(x_1, \ldots, x_n), \ldots, f_n(x_1, \ldots, x_n))} = \\ &= f(\overline{f_1(x_1, \ldots, x_n)}, \ldots, \overline{f_n(x_1, \ldots, x_n)}) = \\ &= f(f_1(\bar{x}_1, \ldots, \bar{x}_n), \ldots, f_n(\bar{x}_1, \ldots, \bar{x}_n)) = F(\bar{x}_1, \ldots, \bar{x}_n),\end{aligned}$$

тобто $F \in K_s$. □

Наслідок 7. *Повна система булевих функцій містить хоча б одну несамодвоїсту функцію.*

Справді, якщо припустити, що всі функції повної системи самодвоїсті, то звідси випливатиме, що будь-яка булева функція також буде самодвоїстою. Але ж це не так, оскільки, наприклад, імплікація є несамодвоїста функція, бо вона на протилежних наборах $(0, 0)$ і $(1, 1)$ приймає однакові значення. Отже, $\longrightarrow \notin K_s$.

Лема 4. *За допомогою підстановки функцій x і $\bar{x}$ у несамодвоїсту булеву функцію можна отримати константу.*

Доведення. Нехай функція $f(x_1, \ldots, x_n)$ несамодвоїста, тобто

$$f(a_1, \ldots, a_n) = f(\bar{a}_1, \ldots, \bar{a}_n) \tag{1.5.3}$$

для деякого двійкового набору $\vec{a} = (a_1, \ldots, a_n)$. За набором $\vec{a}$ побудуємо функції $\varphi_i(x)$, $i = 1, \ldots, n$, таким чином:

$$\varphi_i(x) = \begin{cases} x, & \text{якщо } a_i = 0; \\ \bar{x}, & \text{якщо } a_i = 1. \end{cases} \tag{1.5.4}$$

З (1.5.4) випливає, що $\varphi_i(0) = a_i$, $\varphi_i(1) = \bar{a}_i$ для кожного $i = 1, \ldots, n$. Розглянемо функцію

$$\varphi(x) = f(\varphi_1(x), \ldots, \varphi_n(x)), \tag{1.5.5}$$

тоді матимемо

$$\varphi(0) = f(\varphi_1(0), \ldots, \varphi_n(0)) = f(a_1, \ldots, a_n) = f(\bar{a}_1, \ldots, \bar{a}_n) = f(\varphi_1(1), \ldots, \varphi_n(1)) = \varphi(1),$$

тобто $\varphi(x)$ є константа. □

3. Кажуть, що набір $\vec{a} = (a_1, \ldots, a_n)$ *передує* набору $\vec{b} = (b_1, \ldots, b_n)$ і це позначається $\vec{a} \prec \vec{b}$, якщо $a_i \leqslant b_i$ для всіх $i = 1, \ldots, n$. Булева функція $f(x_1, \ldots, x_n)$ називається *монотонною*, якщо для довільних наборів $\vec{a}$, $\vec{b}$ з того, що $\vec{a} \prec \vec{b}$, випливає $f(\vec{a}) \leqslant f(\vec{b})$. Клас всіх монотонних функцій позначимо через K_m.

Лема 5. *Суперпозиція монотонних булевих функцій є знову монотонна функція, тобто клас K_m замкнений відносно суперпозицій.*

Доведення. Нехай $f, f_1, \ldots, f_n \in K_m$ і F визначається згідно (1.5.1). Якщо $\vec{a} \prec \vec{b}$, то, очевидно, $f_i(\vec{a}) \leqslant f_i(\vec{b})$ для кожного $i = 1, \ldots, n$, тому $(f_1(\vec{a}), \ldots, f_n(\vec{a})) \prec (f_1(\vec{b}), \ldots, f_n(\vec{b}))$, звідки

$$F(\vec{a}) = f(f_1(\vec{a}), \ldots, f_n(\vec{a})) \prec f(f_1(\vec{b}), \ldots, f_n(\vec{b})) = F(\vec{b}),$$

отже, $F \in K_m$. □

Наслідок 8. *Повна система булевих функцій містить хоча б одну немонотонну функцію.*

Справді, якщо припустити, що всі функції повної системи монотонні, то звідси випливатиме, що будь-яка булева функція також буде монотонною. Але ж це не так, оскільки, наприклад, еквівалентність є немонотонна функція, оскільки існують двійкові набори $\vec{a} = (0,0)$ і $\vec{b} = (0,1)$ такі, що $\vec{a} \prec \vec{b}$, але $0 \longleftrightarrow 0 = 1$ і $0 \longleftrightarrow 1 = 0$. Отже, $\longleftrightarrow \notin K_m$.

Лема 6. *За допомогою підстановки у немонотонну функцію констант 0 і 1, а також функції x, можна отримати $\bar{x}$.*

Доведення. Нехай $f(x_1, \ldots, x_n)$ є немонотонна функція. Це означає, що існують такі двійкові набори $\vec{a} = (a_1, \ldots, a_n)$, $\vec{b} = (b_1, \ldots, b_n)$, що $\vec{a} \prec \vec{b}$, але $f(\vec{a}) > f(\vec{b})$. З останнього випливає, що $f(\vec{a}) = 1$, $f(\vec{b}) = 0$. Побудуємо тепер послідовність сусідніх наборів $\vec{a}_0, \vec{a}_1, \vec{a}_2, \ldots, \vec{a}_m$ [10] таких, що $\vec{a} = \vec{a}_0 \prec \vec{a}_1 \prec \vec{a}_2 \prec \ldots \prec \vec{a}_m = \vec{b}$. Оскільки $f(\vec{a}_0) = 1$ і $f(\vec{a}_m) = 0$, то знайдуться такі сусідні набори $\vec{a}_k$ і $\vec{a}_{k+1}$, що $f(\vec{a}_k) = 1$ і $f(\vec{a}_{k+1}) = 0$. Припустимо, що ці набори мають такий вид:

$$\vec{a}_k = (a_1, \ldots, a_{i-1}, 0, a_{i+1}, \ldots, a_n), \quad \vec{a}_{k+1} = (a_1, \ldots, a_{i-1}, 1, a_{i+1}, \ldots, a_n),$$

де $a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n \in \{0,1\}$. Розглянемо функцію

$$g(x) = f(a_1, \ldots, a_{i-1}, x, a_{i+1}, \ldots, a_n).$$

Маємо $g(0) = f(\vec{a}_k) = 1$ і $g(1) = f(\vec{a}_{k+1}) = 0$, тому $g(x) = \bar{x}$. □

4. Булева функція $f(x_1, \ldots, x_n)$ називається *лінійною*, якщо вона подається поліномом Жегалкіна першого степеня, тобто якщо вона має вид:

$$f(x_1, \ldots, x_n) = a_n x_n \oplus a_{n-1} x_{n-1} \oplus \ldots \oplus a_1 x_1 \oplus a_0,$$

де $a_0, a_1, \ldots, a_n \in \{0,1\}$. Клас всіх лінійних булевих функцій позначимо через K_l.

Лема 7. *Суперпозиція лінійних булевих функцій є знову лінійна функція, тобто клас K_l замкнений відносно суперпозицій.*

[10] Двійкові набори $\vec{a}_i$ і $\vec{a}_{i+1}$ називаються *сусідніми*, якщо вони відрізняються рівно однією компонентою, наприклад, $(0,1,0,1,0)$ і $(0,1,0,1,1)$ — сусідні набори.

Доведення леми очевидне.

Наслідок 9. *Повна система булевих функцій містить хоча б одну нелінійну функцію.*

Міркування аналогічні до міркувань аналогічних лем попередній пунктів, тому ми їх опускаємо. Зауважимо лише, що існують нелінійні булеві функції, наприклад, імплікація (див. стор. 27).

Лема 8. *З нелінійної функції $f(x_1, \ldots, x_n)$ констант $0,\ 1$ і функцій x, $\bar{x}$, y суперпозицією можна отримати кон'юнкцію двох логічних змінних.*

Доведення. Оскільки $f(x_1, \ldots, x_n)$ є нелінійна функція, то в поліномі Жегалкіна, який її зображує, існує одночлен, який містить добуток двох логічних змінних, скажімо, x_1x_2. Далі, всі одночлени полінома розіб'ємо на чотири групи:

I — які містять одночасно x_1 і x_2;

II — які містять x_1, але не містять x_2;

III — які не містять x_1, але містять x_2;

IV — які не містять ні x_1, ні x_2.

Функція $f(x_1, \ldots, x_n)$ в результаті такого групування буде мати вигляд:

$$f(x_1, \ldots, x_n) = x_1x_2\Psi_1(x_3, \ldots, x_n) \oplus x_1\Psi_2(x_3, \ldots, x_n) \oplus x_2\Psi_3(x_3, \ldots, x_n) \oplus \Psi_4(x_3, \ldots, x_n).$$

Очевидно, $\Psi_1(x_3, \ldots, x_n) \not\equiv 0$, інакше $f(x_1, \ldots, x_n)$ була б лінійною функцією. Отже, знайдеться такий такий набір значень змінних $x_3, \ldots, x_n$, наприклад, $\vec{a} = (a_3, \ldots, a_n)$, що $\Psi_1(a_3, \ldots, a_n) = 1$. Розглянемо функцію

$$\xi(x_1, x_2) = x_1x_2 \oplus \alpha x_1 \oplus \beta x_2 \oplus \gamma,$$

де $\alpha = \Psi_2(a_3, \ldots, a_n)$, $\beta = \Psi_3(a_3, \ldots, a_n)$, $\gamma = \Psi_4(a_3, \ldots, a_n)$. За допомогою цієї функції побудуємо функцію $F(x, y)$ так:

$$\begin{aligned} F(x,y) &= \xi(x \oplus \beta, y \oplus \alpha) = (x \oplus \beta)(y \oplus \alpha) \oplus \alpha(x \oplus \beta) \oplus \beta(y \oplus \alpha) \oplus \gamma = \\ &= xy \oplus \alpha x \oplus \beta y \oplus \alpha\beta \oplus \alpha x \oplus \alpha\beta \oplus \beta y \oplus \alpha\beta \oplus \gamma = \\ &= xy \oplus (\alpha\beta \oplus \gamma), \end{aligned}$$

тобто $F(x, y) = xy \oplus (\alpha\beta \oplus \gamma)$.

а) Якщо $\alpha\beta \oplus \gamma = 0$, то $F(x, y) = xy$.

б) Якщо $\alpha\beta \oplus \gamma = 1$, то $F(x, y) = xy \oplus 1 = \overline{xy}$,

тому $xy = \overline{F(x, y)}$. Отже, нами побудована кон'юнкція логічних змінних x і y. □

5. Теорема про функціональну повноту.

Теорема 16 (Post E.). *Для того щоб система булевих функцій* $\{f_1, f_2, \ldots, f_n\}$ *була повною, необхідно і достатньо, щоб вона містила:*

- *хоча б одну функцію, яка не зберігає константу* 0*;*
- *хоча б одну функцію, яка не зберігає константу* 1*;*
- *хоча б одну несамодвоїсту функцію;*
- *хоча б одну немонотонну функцію;*
- *хоча б одну нелінійну функцію.*

Доведення. Необхідність умов теореми випливає із наслідків 5, 6, 7, 8, 9. Для доведення достатності припустимо, що виконуються всі п'ять умов теореми і нехай $\{f_0, f_1, f_s, f_m, f_l\}$ є підсистема системи $\{f_1, f_2, \ldots, f_n\}$ така, що f_0 не зберігає 0, f_1 не зберігає 1, f_s — несамодвоїста, f_m — немонотонна, f_l — нелінійна функція.

Розглянемо довільну булеву функцію $f(x_1, \ldots, x_n)$ і покажемо, що її можна подати як суперпозицію функцій $f_0, f_1, f_s, f_m, f_l, x_1, \ldots, x_n$. Перш за все, побудуємо константи 0 і 1. Візьмемо f_0 і x_1, і далі побудуємо функцію $g(x_1) = f_0(x_1, \ldots, x_1)$. Оскільки f_0 не зберігає 0, то $g(0) = f_0(0, \ldots, 0) = 1$, тобто $g(0) = 1$.

а) Якщо $g(1) = 1$, тоді, очевидно, $g(x_1) \equiv 1$, тобто $g(x_1)$ є константа 1. Далі, підставляючи $g(x_1)$ в f_1 отримаємо функцію $\sigma(x_1) = f_1(g(x_1), \ldots, g(x_1))$. Маємо

$$\sigma(0) = f_1(g(0), \ldots, g(0)) = f_1(1, \ldots, 1) = 0,$$
$$\sigma(1) = f_1(g(1), \ldots, g(1)) = f_1(1, \ldots, 1) = 0,$$

тому $\sigma(x_1) \equiv 0$, тобто $\sigma(x_1)$ є константа 0.

б) Якщо ж $g(1) = 0$, то, очевидно, $g(x_1) = \bar{x}_1$. Далі, з f_s і $\bar{x}_1$ за лемою 4 будуємо якусь константу. Іншу константу отримуємо із побудованої в результаті її підстановки в $\bar{x}_1$. Таким чином, константи 0 і 1 нами побудовані.

Потім за лемою 6 з f_m, 0 і 1 будуємо заперечення змінної $\bar{x}_1$. Після цього, за лемою 8 з f_l, 0, 1, $\bar{x}_1$, x_1, x_2 будуємо кон'юнкцію x_1x_2. Відомо, що функція $f(x_1, \ldots, x_n)$ може бути задана формулою, яка записана лише за допомогою $x_1, \ldots, x_n$, заперечення і кон'юнкції. Але оскільки останні операції виражаються через $f_0, f_1, f_s, f_m, f_l, x_1, \ldots, x_n$, то $f(x_1, \ldots, x_n)$ також може бути задана суперпозицією функцій $f_0, f_1, f_s, f_m, f_l, x_1, \ldots, x_n$. Отже, ми показали, що система f_0, f_1, f_s, f_m, f_l повна, а значить система $\{f_1, f_2, \ldots, f_n\}$ також повна. □

Приклад. Дослідити на повноту систему булевих функцій $\{\sim, \longleftrightarrow\}$.

Дослідження даної системи функцій виконаємо за теоремою Поста. Маємо $\bar{0} = 1$. Отже, $\sim$ не зберігає константу 0, тому перша умова теореми 16 виконується. Аналогічно $\bar{1} = 0$, тому $\sim$ не зберігає константу 1. Друга умова теореми 16 виконується також. Далі маємо $0 \longleftrightarrow 0 = 1$ і $1 \longleftrightarrow 1 = 1$, тому $\longleftrightarrow$ є несамодвоїста функція. Отже, третя умова теореми виконується. Оскільки $0 < 1$, але $\bar{0} = 1 > 0 = \bar{1}$, то $\sim$ є функція немонотонна, а тому і четверта умова теореми має місце. І нарешті, $\bar{x} = x \oplus 1$, $x \longleftrightarrow y = x \oplus y \oplus 1$, що говорить про те, що обидві функції лінійні. Таким чином, п'ята умова теореми не має місця. Отже, дана система булевих функцій неповна.

2 Числення висловлень

2.1 Числення висловлень. Теорема дедукції

Поняття про формальну аксіоматичну теорію. Вивідність із гіпотез та її властивості. Аксіоми числення висловлень. Лема $\vdash A \longrightarrow A$. *Теорема дедукції.*

1. Таблиці істинності дозволяють відповісти на багато важливих питань математичної логіки, в тому числі, і на питання: чи буде дана формула тавтологією, протиріччям або ні тим та не іншим, чи випливає вона логічно з іншої формули тощо. Однак більш складні питання математичної логіки, про які піде мова пізніше, вже не можуть бути вирішені за допомогою таблиць істинності. Для їх розв'язання існує інший метод, а саме, *метод формальних теорій.*

Формальна аксіоматична теорія $\mathbf{T}$ вважається визначеною, якщо виконані наступні умови:

1. Задана деяка зчисленна множина символів теорії $\mathbf{T}$ (ця множина називається *алфавітом* теорії $\mathbf{T}$). Скінченні послідовності символів теорії $\mathbf{T}$ називаються *виразами теорії* $\mathbf{T}$.

2. Виділена підмножина виразів теорії $\mathbf{T}$, елементи якої називаються *формулами*. Звичайно існує правило, що дозволяє для деякого виразу визначати, чи є воно формулою чи ні.

3. Виділена деяка підмножина формул теорії $\mathbf{T}$, елементи якої називаються *аксіомами теорії* $\mathbf{T}$. Часто вимагається лише ефективно з'ясувати, чи є дана формула теорії $\mathbf{T}$ аксіомою. чи ні; в цьому випадку теорія $\mathbf{T}$ називається *ефективно аксіоматизованою* або *аксіоматичною теорією.*

4. Задана деяка скінченна множина $\rho_1, \ldots, \rho_n$ відношень між формулами теорії $\mathbf{T}$, які називаються *правилами виведення*. Нехай ρ_i є m_i-арне відношення, $A_1, \ldots, A_{m_i-1}, B$ — формули теорії $\mathbf{T}$ такі, що $(A_1, \ldots, A_{m_i-1}, B) \in \rho_i$, тоді формула B називається *безпосереднім наслідком формул* $A_1, \ldots, A_{m_i-1}$ *за правилом* ρ_i. Досить часто правило виведення записують так:

$$\frac{A_1, \ldots, A_{m_i-1}}{B} \ (\rho_i).$$

Доведенням в теорії $\mathbf{T}$ (або *виведенням*) називається кожна скінченна послідовність $A_1, \ldots, A_n$ формул цієї теорії така, що для довільного $i = 1, \ldots, n$ формула A_i є або аксіома теорії $\mathbf{T}$, або безпосередній наслідок з деяких попередніх формул за одним з правил виведення. Формула A теорії $\mathbf{T}$ називається *теоремою теорії* $\mathbf{T}$ (і це позначається $\vdash_{\mathbf{T}} A$ або просто $\vdash A$), якщо існує таке доведення в теорії $\mathbf{T}$, що A є в ньому останньою формулою. Це доведення називається *доведенням формули* A.

Відмітимо, що для поняття теореми може і не існувати алгоритму, який дозволяє пізнавати за даною формулою, чи існує доведення в теорії $\mathbf{T}$ цієї формули або ні. Теорія, для якої такий алгоритм існує, називається *розв'язною*, в іншому випадку

вона називається *нерозв'язною*.

2. Формула A називається *наслідком в теорії* $\mathbf{T}$ *множини формул* Γ (які називаються *гіпотезами*) тоді і тільки тоді, коли існує така скінченна послідовність формул $A_1, \ldots, A_n$, що A_n є A, і для кожного $i = 1, \ldots, n$ формула A_i є або аксіома, або елемент множини Γ, або безпосереднім наслідком деяких формул, що їй передують, за одним із правил виведення. Твердження "A є наслідок Γ"символічно позначається через $\Gamma \vdash A$. Якщо ж $\Gamma = \{B_1, \ldots, B_n\}$, то пишемо $B_1, \ldots, B_n \vdash A$, якщо ж $\Gamma = \varnothing$, то замість $\varnothing \vdash A$ пишемо $\vdash A$. Останнє, як не важко бачити, означає, що A є теорема теорії $\mathbf{T}$.

Відмітимо деякі властивості вивідності із гіпотез:

1. Якщо $\Gamma \subset \Delta$ і $\Gamma \vdash A$, то $\Delta \vdash A$.
2. $\Gamma \vdash A$ тоді і тільки тоді, коли існує скінченна підмножина $\Delta \subset \Gamma$ така, що $\Delta \vdash A$.
3. Якщо $\Delta \vdash A$ і $\Gamma \vdash B$ для довільного $B \in \Delta$, то $\Gamma \vdash A$.

3. Розглянемо тепер формальну аксіоматичну теорію $\mathbf{L}$ для *числення висловлень*.

1. *Символами* $\mathbf{L}$ є $\sim$, $\longrightarrow$, (,)і літери $p_1, q_1, r_1, s_1, p_2, q_2, r_2, s_2, \ldots$ Символи $\sim$ і $\longrightarrow$ називаються *логічними зв'язками*, а літери p_i, q_i, r_i, s_i — *логічними змінними*.
2. *Формули* $\mathbf{L}$ визначаються так:
   (a) Всі логічні змінні є формули.
   (b) Якщо A і B — формули, то вирази $(\sim A)$ і $(A \longrightarrow B)$ — також формули.
3. Які б не були формули A, B, C теорії $\mathbf{L}$ наступні формули є *аксіоми* $\mathbf{L}$:[11]

   $\mathbf{A}_1$: $(A \longrightarrow (B \longrightarrow A))$;
   $\mathbf{A}_2$: $((A \longrightarrow (B \longrightarrow C)) \longrightarrow ((A \longrightarrow B) \longrightarrow (A \longrightarrow C)))$;
   $\mathbf{A}_3$: $(((\sim B) \longrightarrow (\sim A)) \longrightarrow (((\sim B) \longrightarrow A) \longrightarrow B))$.
4. Єдиним правилом виведення є правило *modus ponens*: B є безпосередній наслідок A і $A \longrightarrow B$, тобто

$$\frac{A,\ A \longrightarrow B}{B}\ \text{(modus ponens або скорочено MP)}.$$

Відмітимо, що зовнішні дужки ми будемо опускати, якщо це не буде приводити до непорозумінь. Наша мета: *побудувати теорію* $\mathbf{L}$ *таким чином, щоб клас всіх її теорем співпадав з класом всіх тавтологій*.

Інші логічні зв'язки введемо за допомогою таких означень:

$\mathbf{D}_1$: $(A \wedge B)$ означає $\sim (A \longrightarrow (\sim B))$;

$\mathbf{D}_2$: $(A \vee B)$ означає $(\sim A) \longrightarrow B$;

[11] Зауважимо, що вирази $\mathbf{A}_1$, $\mathbf{A}_2$ і $\mathbf{A}_3$ є, так звані, схеми аксіом, кожна з яких визначає нескінченну множину аксіом, тобто підставляючи в дані схеми конкретні формули, ми будемо кожний раз отримувати конкретні аксіоми.

$\mathbf{D}_3$: $(A \longleftrightarrow B)$ означає $(A \longrightarrow B) \wedge (B \longrightarrow A)$.

Лема 9. *Для довільної формули A теорії* $\mathbf{L}$ *має місце* $\vdash A \longrightarrow A$.

Доведення.

1. $(A \longrightarrow ((A \longrightarrow A) \longrightarrow A)) \longrightarrow ((A \longrightarrow (A \longrightarrow A)) \longrightarrow (A \longrightarrow A))$ (підстановка в схему аксіом $\mathbf{A}_2$)
2. $A \longrightarrow ((A \longrightarrow A) \longrightarrow A)$ (підстановка в схему $\mathbf{A}_1$)
3. $(A \longrightarrow (A \longrightarrow A)) \longrightarrow (A \longrightarrow A)$ (з 1 і 2 по MP)
4. $A \longrightarrow (A \longrightarrow A)$ (схема аксіом $\mathbf{A}_1$)
5. $A \longrightarrow A$ (з 3 і 4 по MP) □

4. В математичних міркуваннях часто деяке твердження B доводять в припущенні справедливості іншого твердження A, після чого роблять висновок, що справедливе твердження "якщо A, то B". Для системи $\mathbf{L}$ цей засіб обгрунтовується наступною теоремою.

Теорема 17 (Метатеорема дедукції). *Якщо Γ — множина формул теорії* $\mathbf{L}$*, A і B — формули* $\mathbf{L}$ *і $\Gamma, A \vdash B$, то $\Gamma \vdash A \longrightarrow B$.*

Доведення. Нехай $B_1, \ldots, B_n$ є доведення формули B з $\Gamma \cup \{A\}$, де $B_n = B$. За допомогою метода математичної індукції доведемо, що $\Gamma \vdash A \longrightarrow B_i$ для кожного $i = 1, \ldots, n$. Для формули B_1 можливі такі випадки: а) B_1 — аксіома, б) $B_1 \in \Gamma$, в) B_1 є формула A. За схемою $\mathbf{A}_1$ ми можемо записати $\vdash B_1 \longrightarrow (A \longrightarrow B_1)$, тому у випадках а) і б) за MP отримуємо $\Gamma \vdash A \longrightarrow B_1$. У випадку ж в) за лемою 9 маємо $\vdash A \longrightarrow B_1$, тому $\Gamma \vdash A \longrightarrow B_1$. Отже, при $i = 1$ теорема справедлива.

Припустимо, що теорема справедлива для довільного $k < i$, тобто $\Gamma \vdash A \longrightarrow B_k$ для всіх $k < i$. Для формули B_i є чотири можливості: а) B_i — аксіома; б) $B_i \in \Gamma$; в) B_i є формула A; г) B_i виводиться за MP з B_j і $B_m = B_j \longrightarrow B_i$, де $j < i$ і $m < i$. У випадках а), б) і в) за допомогою таких же міркувань, як і при $i = 1$, показуємо $\Gamma \vdash A \longrightarrow B_i$.У випадку ж г) за схемою $\mathbf{A}_2$ записуємо, що

$$\vdash (A \longrightarrow (B_j \longrightarrow B_i)) \longrightarrow ((A \longrightarrow B_j) \longrightarrow (A \longrightarrow B_i)).$$

Але згідно індуктивного припущення $\Gamma \vdash A \longrightarrow B_j$ і $\Gamma \vdash A \longrightarrow (B_j \longrightarrow B_i)$. Отже, по MP отримуємо спочатку $\Gamma \vdash (A \longrightarrow B_j) \longrightarrow (A \longrightarrow B_i)$, а далі знову по MP отримуємо $\Gamma \vdash A \longrightarrow B_i$. Отже, згідно методу математичної індукції $\Gamma \vdash A \longrightarrow B_i$ для всіх $i = 1, \ldots, n$, тому $\Gamma \vdash A \longrightarrow B_n$, що означає $\Gamma \vdash A \longrightarrow B$. □

Наслідок 10. *Якщо $A \vdash B$, то $\vdash A \longrightarrow B$.*

Приклад 1. Користуючись теоремою дедукції довести, що

$$A \longrightarrow B, B \longrightarrow C \vdash A \longrightarrow C. \tag{2.1.1}$$

Доведемо спочатку, що

$$A \longrightarrow B, B \longrightarrow C, A \vdash C. \tag{2.1.2}$$

Отже, маємо:

1. A (гіпотеза)
2. $A \longrightarrow B$ (гіпотеза)
3. B (MP, 1, 2)
4. $B \longrightarrow C$ (гіпотеза)
5. C (MP, 3, 4)

Таким чином, ми довели (2.1.2). Застосувавши тепер до (2.1.2) теорему дедукції, отримаємо (2.1.1), що і треба було довести. □

Приклад 2. Користуючись теоремою дедукції довести, що

$$A \longrightarrow (B \longrightarrow C), B \vdash A \longrightarrow C. \tag{2.1.3}$$

Доведемо спочатку, що

$$A \longrightarrow (B \longrightarrow C), B, A \vdash C. \tag{2.1.4}$$

Отже, маємо:

1. A (гіпотеза)
2. $A \longrightarrow (B \longrightarrow C)$ (гіпотеза)
3. $B \longrightarrow C$ (MP, 1, 2)
4. B (гіпотеза)
5. C (MP, 3, 4)

Таким чином, ми довели (2.1.4). Застосувавши тепер до (2.1.4) теорему дедукції, отримаємо (2.1.3), що і треба було довести. □

2.2 Повнота, несуперечність і незалежність аксіом числення висловлень

Деякі теореми числення висловлень. Теорема про повноту. Несуперечність числення висловлень. Незалежність аксіом числення висловлень. Інші аксіоматики числення висловлень.

1. Доведемо деякі формальні теореми числення висловлень, які нам будуть потрібні далі.

Лема 10. *Для довільних формул A і B наступні формули є теоремами теорії* **L**:

(a) $\sim\sim B \longrightarrow B$;
(b) $B \longrightarrow \sim\sim B$;
(c) $\sim A \longrightarrow (A \longrightarrow B)$;
(d) $(\sim B \longrightarrow \sim A) \longrightarrow (A \longrightarrow B)$;
(e) $(A \longrightarrow B) \longrightarrow (\sim B \longrightarrow \sim A)$;
(f) $A \longrightarrow (\sim B \longrightarrow \sim (A \longrightarrow B))$;
(g) $(A \longrightarrow B) \longrightarrow ((\sim A \longrightarrow B) \longrightarrow B)$.

Доведення.

(a) $\vdash \sim\sim B \longrightarrow B$.

1. $(\sim B \longrightarrow \sim\sim B) \longrightarrow ((\sim B \longrightarrow \sim B) \longrightarrow B)$	(схема аксіом $\mathbf{A}_3$)
2. $\sim B \longrightarrow \sim B$	(лема 9)
3. $(\sim B \longrightarrow \sim\sim B) \longrightarrow B$	(1, 2, (2.1.3))
4. $\sim\sim B \longrightarrow (\sim B \longrightarrow \sim\sim B)$	(схема аксіом $\mathbf{A}_1$)
5. $\sim\sim B \longrightarrow B$	(3, 4, (2.1.1))

(b) $\vdash B \longrightarrow \sim\sim B$.

1. $(\sim\sim\sim B \longrightarrow \sim B) \longrightarrow ((\sim\sim\sim B \longrightarrow B) \longrightarrow \sim\sim B)$	(схема аксіом $\mathbf{A}_3$)
2. $\sim\sim\sim B \longrightarrow \sim B$	(пункт (a), доведений вище)
3. $(\sim\sim\sim B \longrightarrow B) \longrightarrow \sim\sim B$	(1, 2 MP)
4. $B \longrightarrow (\sim\sim\sim B \longrightarrow B)$	(схема аксіом $\mathbf{A}_1$)
5. $B \longrightarrow \sim\sim B$	(3, 4, (2.1.1))

(c) $\vdash \sim A \longrightarrow (A \longrightarrow B)$.

1. $\sim A$	(гіпотеза)
2. A	(гіпотеза)
3. $A \longrightarrow (\sim B \longrightarrow A)$	(схема аксіом $\mathbf{A}_1$)
4. $\sim A \longrightarrow (\sim B \longrightarrow \sim A)$	(схема аксіом $\mathbf{A}_1$)
5. $\sim B \longrightarrow A$	(2, 3, MP)
6. $\sim B \longrightarrow \sim A$	(1, 4, MP)
7. $(\sim B \longrightarrow \sim A) \longrightarrow ((\sim B \longrightarrow A) \longrightarrow A)$	(схема аксіом $\mathbf{A}_3$)
8. $(\sim B \longrightarrow A) \longrightarrow B$	(6, 7, MP)
9. B	(5, 8, MP)

Отже, згідно 1 – 9, $\sim A, A \vdash B$. Тому, за теоремою дедукції, $\sim A \vdash A \longrightarrow B$ і, за цією теоремою, $\vdash \sim A \longrightarrow (A \longrightarrow B)$.

(d) $\vdash (\sim B \longrightarrow \sim A) \longrightarrow (A \longrightarrow B)$.

1. $\sim B \longrightarrow \sim A$	(гіпотеза)
2. A	(гіпотеза)
3. $(\sim B \longrightarrow \sim A) \longrightarrow ((\sim B \longrightarrow A) \longrightarrow B)$	(схема аксіом $\mathbf{A}_3$)
4. $A \longrightarrow (\sim B \longrightarrow A)$	(схема аксіом $\mathbf{A}_1$)
5. $(\sim B \longrightarrow A) \longrightarrow B$	(1, 3, MP)
6. $A \longrightarrow B$	(4, 5, (2.1.1))
7. B	(2, 6, MP)

В силу 1 – 7, $\sim B \longrightarrow \sim A, A \vdash B$, після чого, двічі застосовуючи теорему дедукції, отримаємо потрібний результат.

(e) $\vdash (A \longrightarrow B) \longrightarrow (\sim B \longrightarrow \sim A)$.

1. $A \longrightarrow B$	(гіпотеза)
2. $\sim\sim A \longrightarrow A$	(пункт (a))
3. $\sim\sim A \longrightarrow B$	(1, 2, (2.1.1S))
4. $B \longrightarrow \sim\sim B$	(пункт (b))
5. $\sim\sim A \longrightarrow \sim\sim B$	(3, 4, (2.1.1))
6. $(\sim\sim A \longrightarrow \sim\sim B) \longrightarrow (\sim B \longrightarrow \sim A)$	(пункт (d))
7. $(\sim B \longrightarrow \sim A))$	(5, 6, MP)

В силу 1 – 7, $A \longrightarrow B \vdash \sim B \longrightarrow \sim A$, звідки (e) отримується за теоремою дедукції.

(f) $\vdash A \longrightarrow (\sim B \longrightarrow \sim (A \longrightarrow B))$.

Очевидно, $A, A \longrightarrow B \vdash B$. Застосувавши двічі теорему дедукції, отримуємо $\vdash A \longrightarrow ((A \longrightarrow B) \longrightarrow B)$. Згідно пункту (e) маємо

$$\vdash ((A \longrightarrow B) \longrightarrow B) \longrightarrow (\sim B \longrightarrow \sim (A \longrightarrow B)).$$

Нарешті, застосувавши (2.1.1), отримуємо

$$\vdash A \longrightarrow (\sim B \longrightarrow \sim (A \longrightarrow B)).$$

(g) $\vdash (A \longrightarrow B) \longrightarrow ((\sim A \longrightarrow B) \longrightarrow B)$.

1. $A \longrightarrow B$	(гіпотеза)
2. $\sim A \longrightarrow B$	(гіпотеза)
3. $(A \longrightarrow B) \longrightarrow (\sim B \longrightarrow \sim A)$	(пункт (e))

4. $\sim B \longrightarrow \sim A$ (1, 3, MP)

5. $(\sim A \longrightarrow B) \longrightarrow (\sim B \longrightarrow \sim\sim A)$ (пункт (e))

6. $\sim B \longrightarrow \sim\sim A$ (2, 5, MP)

7. $(\sim B \longrightarrow \sim\sim A) \longrightarrow ((\sim B \longrightarrow \sim A) \longrightarrow B)$ (схема аксіом $\mathbf{A}_3$)

8. $(\sim B \longrightarrow \sim A) \longrightarrow B$ (6, 7, MP)

9. B (4, 8, MP)

Отже, $A \longrightarrow B, \sim A \longrightarrow B \vdash B$. Застосовуючи далі два рази теорему дедукції отримуємо (g).

Твердження 1. *Кожна теорема числення висловлень є тавтологія.*

Справді, легко бачити, що кожна з аксіом числення висловлень є тавтологія. Відомо, що MP, застосований до тавтологій, знову приводить до тавтології, звідки випливає справедливість твердження.

Лема 11. *Нехай A є формула числення висловлень, а $p_1, \ldots, p_k$ — логічні змінні, які входять в A. Нехай заданий деякий розподіл істинностних значень для $p_1, \ldots, p_k$. Нехай далі p'_i означає p_i, якщо $|p_i| = 1$, і p'_i означає $\sim p_i$, якщо $|p_i| = 0$. Аналогічно, A' є A при $|A| = 1$ і A' є $\sim A$ при $|A| = 0$. Тоді справедливе*

$$p'_1, \ldots, p'_k \vdash A'. \tag{2.2.1}$$

Доведення. Доведення проводиться індукцією по числу n входжень в A логічних зв'язок. Якщо $n = 0$, то A являє собою деяку логічну змінну, скажімо, p_1. В цьому випадку твердження леми зводиться до $p_1 \vdash p_1$, або до $\sim p_1 \vdash \sim p_1$, що є справедливим в силу леми 9. Отже, при $n = 0$ твердження (2.2.1) має місце. Припустимо, що (2.2.1) справедливе для кожного $i < n$.

Випадок 1. Формула A має вигляд заперечення $\sim B$, де число входжень логічних зв'язок в формулу B меньше n.

а) Нехай при даному розподілі істинностних значень $|B| = 1$, тоді $|A| = 0$, тому $B' = B$ і $A' = \sim A$. Згідно припущення $p'_1, \ldots, p'_k \vdash B$, але за лемою 10 (b) $\vdash B \longrightarrow \sim\sim B$, тому по MP отримуємо $p'_1, \ldots, p'_k \vdash \sim\sim B$, але $\sim\sim B = \sim A = A'$, звідки $p'_1, \ldots, p'_k \vdash A'$.

б) Нехай тепер $|B| = 0$, тоді $|A| = 1$, звідки $B' = \sim B$ і $A' = A$. За припущенням $p'_1, \ldots, p'_k \vdash B'$, тобто $p'_1, \ldots, p'_k \vdash \sim B$, але $\sim B = A = A'$, тому $p'_1, \ldots, p'_k \vdash A'$.

Випадок 2. Нехай A має вид $B \longrightarrow C$, тоді число входжень логічних зв'язок в B і C меньше, ніж в A, тому за припущенням $p'_1, \ldots, p'_k \vdash B'$ і $p'_1, \ldots, p'_k \vdash C'$.

а) Якщо $|B| = 0$, то $|A| = 1$, отже, $B' = \sim B$, $A' = A$. Отже, $p'_1, \ldots, p'_k \vdash \sim B$, але за лемою 10 (c) $\vdash \sim B \longrightarrow (B \longrightarrow C)$, тому по MP $p'_1, \ldots, p'_k \vdash B \longrightarrow C$, тобто $p'_1, \ldots, p'_k \vdash A'$.

б) Якщо $|C| = 1$, то $|A| = 1$, звідки $C' = C$ і $A' = A$. Маємо, $p'_1, \ldots, p'_k \vdash C$, а за схемою $\mathbf{A}_1 \vdash C \longrightarrow (B \longrightarrow C)$, тому по MP $p'_1, \ldots, p'_k \vdash B \longrightarrow C$, тобто $p'_1, \ldots, p'_k \vdash A'$.

в) Якщо ж $|B| = 1$, $|C| = 0$, то $|A| = 0$, тому $B' = B$, $C' = \sim C$, $A' = \sim A = \sim (B \longrightarrow C)$. За припущенням $p'_1, \ldots, p'_k \vdash B$ і $p'_1, \ldots, p'_k \vdash \sim C$. За лемою 10 (f) маємо $\vdash B \longrightarrow (\sim C \longrightarrow \sim (B \longrightarrow C))$, звідки по MP $p'_1, \ldots, p'_k \vdash \sim C \longrightarrow \sim (B \longrightarrow C)$ і далі по MP $p'_1, \ldots, p'_k \vdash \sim (B \longrightarrow C)$, тобто $p'_1, \ldots, p'_k \vdash A'$.

Таким чином, (2.2.1) справедливе для n, тому за методом математичної індукції (2.2.1) виконується для довільного натурального n. □

Теорема 18 (теорема про повноту). *Якщо формула A числення висловлень є тавтологією, то вона є теоремою числення висловлень.*

Доведення. Нехай формула A є тавтологія, $p'_1, \ldots, p'_k$ — логічні змінні, що входять в A. При довільному розподілі істинностних значень змінних $p'_1, \ldots, p'_k$ за лемою 2.2.1 маємо $p'_1, \ldots, p'_k \vdash A$, оскільки $A' = A$. Отже, коли $|p_k| = 1$ маємо $p'_1, \ldots, p'_{k-1}, p_k \vdash A$, а коли $|p_k| = 0$, то $p'_1, \ldots, p'_{k-1}, \sim p_k \vdash A$. Звідси за теоремою дедукції отримуємо $p'_1, \ldots, p'_{k-1} \vdash p_k \longrightarrow A$ і $p'_1, \ldots, p'_{k-1} \vdash \sim p_k \longrightarrow A$. За лемою 10 (g) маємо За лемою 10 (f) маємо

$$\vdash (p_k \longrightarrow A) \longrightarrow ((\sim p_k \longrightarrow A) \longrightarrow A),$$

тому по MP виводимо $p'_1, \ldots, p'_{k-1} \vdash (\sim p_k \longrightarrow A) \longrightarrow A$, звідки знову за MP отримуємо $p'_1, \ldots, p'_{k-1} \vdash A$. Продовжуючи аналогічні міркування далі, ми отримаємо в результаті $\vdash A$, що і треба було довести. □

Наслідок 11. *Якщо вираз B містить знаки $\sim, \longrightarrow, \wedge, \vee, \longleftrightarrow$ і є скороченням для деякої формули A числення висловлень, то B є тавтологією тоді і тільки тоді, коли A є теорема числення висловлень.*

2. Має місце наступна теорема про несуперечність числення висловлень.

Теорема 19. *Числення висловлень несуперечлива формальна теорія, тобто не існує такої формули A, щоб A і $\sim A$ одночасно були теоремами числення висловлень.*

Справді, згідно твердженню 1 кожна теорема числення висловлень є тавтологія. Заперечення тавтології не є тавтологією. Отже, ні для жодної формули A неможливо, щоб A і $\sim A$ були теоремами числення висловлень.

3. Розглянемо тепер питання про незалежність аксіом числення висловлень.

Означення 5. *Підмножина X множини всіх аксіом називається незалежною, якщо деяка формула з X не може бути виведена за допомогою правил виведення з аксіом, що не входять в X.*

Теорема 20. *Кожна із схем аксіом $\mathbf{A}_1 - \mathbf{A}_3$ визначає незалежну множину аксіом.*

Доведення. а) *Незалежність* $\mathbf{A}_1$. На множині з трьох елементів $\{0, 1, 2\}$ задамо операції $\sim, \longrightarrow$ за допомогою таких таблиць:

A	$\sim A$
0	1
1	1
2	0

A	B	$A \longrightarrow B$
0	0	0
1	0	2
2	0	0
0	1	2
1	1	2
2	1	0
0	2	2
1	2	0
2	2	0

Формула A, яка приймає значення 0 для довільних значень змінних називається *виділеною*. Modus ponens зберігає властивість "бути виділеною формулою", тобто якщо A, $A \longrightarrow B$ є виділені формули, то і B є виділена формула. Неважко перевірити, що кожна аксіома, яка отримується за схемою $\mathbf{A}_2$ і $\mathbf{A}_3$, також буде виділеною. Отже, виділеною буде і кожна формула, яка виводиться із схем $\mathbf{A}_2$ і $\mathbf{A}_3$ за допомогою modus ponens. Однак формула $p_1 \longrightarrow (p_2 \longrightarrow p_1)$, яка являє собою частинний випадок $\mathbf{A}_1$ не є виділеною, оскільки вона приймає значення 2 при $|p_1| = 1$ і $|p_2| = 2$.

б) *Незалежність* $\mathbf{A}_2$. На множині з трьох елементів $\{0,1,2\}$ задамо операції $\sim, \longrightarrow$ за допомогою таких таблиць:

A	$\sim A$
0	1
1	0
2	1

A	B	$A \longrightarrow B$
0	0	0
1	0	0
2	0	0
0	1	2
1	1	2
2	1	0
0	2	1
1	2	0
2	2	0

Формула A, яка приймає значення 0 для довільних значень змінних називається *гротескною*. Modus ponens зберігає властивість "бути гротескною формулою", тобто якщо A, $A \longrightarrow B$ є гротескні формули, то і B є гротескна формула. Неважко перевірити, що кожна аксіома, яка отримується за схемою $\mathbf{A}_1$ і $\mathbf{A}_3$, також буде гротескною. Отже, гротескною буде і кожна формула, яка виводиться із схем $\mathbf{A}_1$ і $\mathbf{A}_3$ за допомогою modus ponens. Однак частинний випадок схеми $\mathbf{A}_2$

$$(p_1 \longrightarrow (p_2 \longrightarrow p_3)) \longrightarrow ((p_1 \longrightarrow p_2) \longrightarrow (p_1 \longrightarrow p_3))$$

не є гротескною, оскільки при $|p_1| = 0$, $|p_2| = 0$ і $|p_3| = 1$ формула приймає значення 2.

в) *Незалежність* $\mathbf{A}_3$. Для довільної формули A через $h(A)$ позначимо формулу, яка отримується з A витиранням всіх входжень знака заперечення. Для кожного часткового випадка A схем $\mathbf{A}_1$ і $\mathbf{A}_2$ формула $h(A)$ є тавтологія. Modus ponens зберігає властивість "мати в якості $h(A)$ тавтологію", тобто якщо $h(A)$ і $h(A \longrightarrow B)$ є тавтології, то $h(B)$ — тавтологія, оскільки $h(A \longrightarrow B)$ співпадає з формулою $h(A) \longrightarrow h(B)$. Отже, кожна формула A, яка виводиться з схем $\mathbf{A}_1$ і $\mathbf{A}_2$ за допомогою modus ponens в якості $h(A)$ має тавтологію. Але формула $h((\sim p_1 \longrightarrow\sim p_1) \longrightarrow ((\sim p_1 \longrightarrow p_1) \longrightarrow p_1))$ співпадає з формулою $(p_1 \longrightarrow p_1) \longrightarrow ((p_1 \longrightarrow p_1) \longrightarrow p_1)$, а вона не є тавтологією. Таким чином, аксіома $(\sim p_1 \longrightarrow\sim p_1) \longrightarrow ((\sim p_1 \longrightarrow p_1) \longrightarrow p_1)$, яка є частковим випадком $\mathbf{A}_3$, не виводиться з $\mathbf{A}_1$ і $\mathbf{A}_2$ за допомогою modus ponens. □

4. Для числення висловлень можуть бути побудовані аксіоматизації з однією лише схемою аксіом. Так наприклад, якщо за логічні зв'язки взяти $\sim$ і $\longrightarrow$, то при єдиному правилі виведення modus ponens достатньою виявляється схема аксіом:

$$(((A \longrightarrow B) \longrightarrow (\sim C \longrightarrow\sim D)) \longrightarrow E) \longrightarrow ((E \longrightarrow A) \longrightarrow (D \longrightarrow A))$$

(Мередіг [1953]).

Іншим прикладом такого ж типу може слугувати система Нікода [1917], в якій використовується лише одна логічна зв'язка "штрих Шеффера", одне правило виведення, згідно якого формула C випливає з формул A і $A \,|\, (B \,|\, C)$, і є одна схема аксіом

$$(A \,|\, (B \,|\, C)) \,|\, ((D \,|\, (D \,|\, D)) \,|\, ((E \,|\, B) \,|\, ((A \,|\, E) \,|\, (A \,|\, E)))).$$

3 Логіка предикатів

3.1 Предикати і квантори

Предикат, область істинності предиката. Логічні функції. Операції над предикатами. Квантори, вільні і зв'язані змінні. Формули логіки предикатів. Істинностні значення формул логіки предикатів. Рівносильність формул логіки предикатів. Випереджена нормальна форма.

1. Нехай $M_1, M_2, \ldots, M_n$ є деякі множини, $x_1, x_2, \ldots, x_n$ — деякі змінні, які набувають значень відповідно з даних множин, тоді *кожне стверджувальне речення, яке містить дані змінні і стає висловленням при кожній заміні їх елементами з відповідних множин, називається n-місним предикатом.* Наприклад, над множиною натуральних чисел речення "x — просте число"є одномісний предикат, а "x кохає y"є двомісний предикат на множині людей. Предикати ми будемо позначати таким чином: $P(x_1, x_2, \ldots, x_n)$, $R(x)$ або $Q(x, y)$. Змінні $x_1, x_2, \ldots, x_n$ будемо називати *предметними змінними.*

Нехай далі $P(x_1, \ldots, x_n)$ є n-місний предикат, визначений на множинах $M_1, \ldots, M_n$, тоді *логічною функцією*, що відповідає даному предикату, називається функція

$$\lambda_P \colon M_1 \times \cdots \times M_n \to \{0, 1\},$$

де для довільних $a_1 \in M_1, \ldots, a_n \in M_n$ за означенням маємо

$$\lambda_P(a_1, \ldots, a_n) = |P(a_1, \ldots, a_n)|.$$

Часто поняття логічної функції ототожнюється з поняттям предиката. Це пояснюється тим, що в математичній логіці нас не цікавить змістовна суть предиката, нам важливо знати яке істинностне значення ставиться у відповідність за допомогою даного предиката тій чи іншій послідовності елементів.

Для n-місного предиката $P(x_1, \ldots, x_n)$, де $x_i \in M_i$, $i = 1, \ldots, n$, через D_P будемо позначати його *область істинності*, яка визначається таким чином:

$$D_P = \{(a_1, \ldots, a_n) \,|\, |P(a_1, \ldots, a_n)| = 1\} \subset M_1 \times \cdots \times M_n.$$

Наприклад, якщо $P(x, y)$ означає предикат $x^2 + y^2 \leqslant 1$ на множині дійсних чисел $\mathbb{R}$, то $D_P = \{(x, y) \in \mathbb{R} \times \mathbb{R} \,|\, x^2 + y^2 \leqslant 1\}$, тобто D_P є круг радіуса 1 з центром в початку координат.

Оскільки предикати для конкретних значень предметних змінних приймають одне із значень "істина"або "хиба", то їх можна зв'язувати логічними операціями. Наприклад, якщо A є висловлення, $P(x)$, $Q(y, z)$ — предикати, то $A \vee P(x)$ — одномісний предикат, $P(x) \longrightarrow\sim Q(y, z)$ є трьохмісний предикат тощо. В зв'язку з цим виникає задача знаходження області істинності предикатів, отримуваних за допомогою логічних операцій, якщо відомі області істинності вихідних предикатів. Справедливе таке твердження:

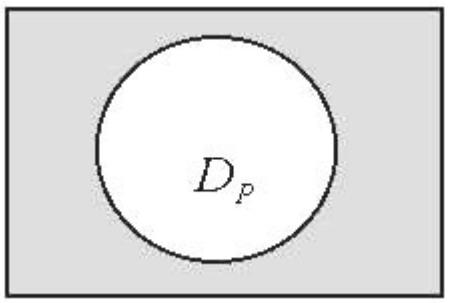
Твердження 2. *Якщо $P(x)$ і $Q(x)$ є предикати над множиною M, то*

$$D_{\sim P} = M \setminus D_P,$$

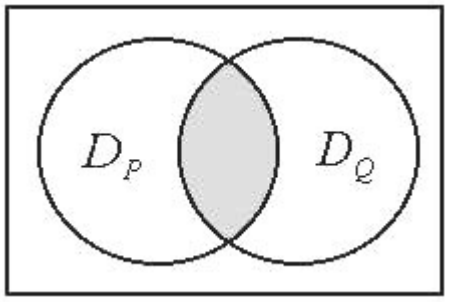
$$D_{P\wedge Q} = D_P \cap D_Q,$$
$$D_{P\vee Q} = D_P \cup D_Q,$$
$$D_{P\longrightarrow Q} = (M \setminus D_P) \cup D_Q,$$
$$D_{P\longleftrightarrow Q} = (D_P \cap D_Q) \cup ((M \setminus D_P) \cap (M \setminus D_Q)).$$

Доведення. Справді, $a \in D_{\sim P}$ означає $|\sim P(a)| = 1$, тобто $|P(a)| = 0$, що означає $a \notin D_{\sim P}$. Отже, ми довели першу рівність. Нехай тепер $a \in D_{P\wedge Q}$, тобто $|(P\wedge Q)(a)| = 1$. Останнє означає, що $|P(a) \wedge Q(a)| = 1$, тому $|P(a)| = 1$ і $|Q(a)| = 1$, тобто $a \in D_P$ і $a \in D_Q$, тому $a \in D_P \cap D_Q$. Друга рівність доведена. Третя доводиться аналогічно. Далі маємо $D_{P\longrightarrow Q} = D_{\sim P\vee Q} = D_{\sim P} \cup D_Q = (M \setminus D_P) \cup D_Q$. Остання рівність доводиться аналогічно. □

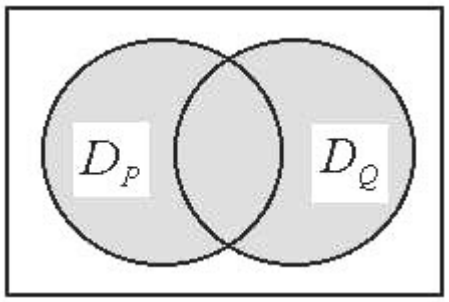
Розглянуті області істинності добре ілюструються нижче діаграмами Ейлера:



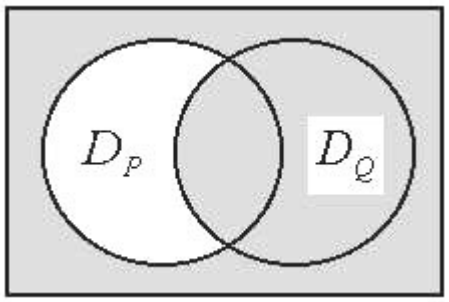
$D_{\sim P}$



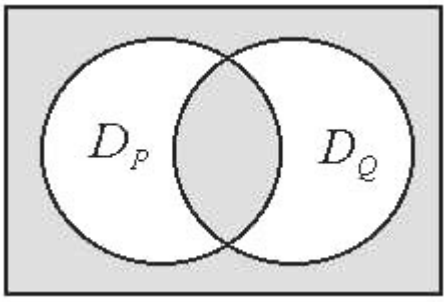
$D_{P\wedge Q}$



$D_{P\vee Q}$



$D_{P\longrightarrow Q}$



$D_{P\longleftrightarrow Q}$

2. Нехай $P(x)$ є деякий предикат над множиною M, тоді під виразом

$$(\forall x)P(x) \tag{3.1.1}$$

ми будемо розуміти *висловлення*, яке істинне, коли $P(x)$ істинне для кожного елемента з M, і хибне в протилежному випадку. Це висловлення вже не залежить від x. Відповідний йому словесний вираз буде такий: *"Для всіх x справедливе $P(x)$"*. Символ $(\forall x)$ називається *квантором загальності*. З даного означення випливає, що у випадку необхідності вираз (3.1.1) можна розглядати як скорочений запис кон'юнкції $\bigwedge\limits_{a\in M} P(a)$. Якщо $M = \{a_1, a_2, \ldots, a_n\}$, то, очевидно, така кон'юнкція має вигляд $P(a_1)\wedge P(a_2)\wedge\ldots\wedge P(a_n)$.

Далі, під виразом

$$(\exists x)P(x) \tag{3.1.2}$$

ми будемо розуміти *висловлення*, яке істинне, коли існує елемент x з M такий, що $P(x)$ істинне. Вираз (3.1.2) словесно читається так: *"Існує x такий, що $P(x)$ справедливе"*. Знак $(\exists x)$ називається *квантором існування*. Відмітимо, що (3.1.2) можна розглядати, як скорочення запису $\bigvee\limits_{a\in M} P(a)$ або ж $P(a_1)\vee P(a_2)\vee\ldots\vee P(a_n)$ при умові, що множина M скінченна, тобто $M = \{a_1, a_2, \ldots, a_n\}$.

За допомогою логічних операцій і кванторів з висловлень та предикатів можна утворювати *формули логіки предикатів*. Наприклад,

$$(\forall x)(A \longrightarrow (\exists y) \sim B(x, y)) \vee (C \longleftrightarrow (\forall z)(\exists y)D(y, z))$$

є формула логіки предикатів. Ми бачимо, що одні предметні змінні знаходяться в області дії квантора, а інші ні. Перші називаються *зв'язаними предметними змінними*, а інші — *вільними*. Наприклад, у формулі $(\forall x)(P(x) \longrightarrow (\exists y)Q(y, z))$ змінні x, y — зв'язані, а z — вільна.

3. Нехай D є деяка множина, яка далі буде називатися *полем*, A — формула логіки предикатів. Відомо, що A може мати змінні трьох видів, а саме, логічні, предметні і предикатні. Логічні змінні, як відомо, приймають значення "істина", "хиба", предметні приймають значення з поля D, а предикатні приймають значення з множини всіх логічних функцій над полем D. *Приписуванням для формули A над полем D* називається кожна відповідність, яка приписує кожній n-місній предикатній змінній n-місну логічну функцію над D, кожній вільній предметній змінній — деякий елемент поля D, кожній логічній змінній — значення "істина"або "хиба". Відмітимо, що логічна функція предиката і його область істинності (яка є відношенням) однозначно визначають одна-одну, тому часто n-місній предикатній змінній приписується просто n-арне відношення.

Як приклад розглянемо питання про приписування істинностних значень для формули

$$(\forall x)(P(x) \longrightarrow Q) \vee (Q \wedge P(y))$$

над полем $D = \{a, b\}$. Згідно означення приписування одномісній предикатній змінній $P(x)$ ставиться у відповідність одна з чотирьох логічних функцій:

x	$\lambda_1(x)$	$\lambda_2(x)$	$\lambda_3(x)$	$\lambda_4(x)$
a	1	1	0	0
b	1	0	1	0

Логічній змінній Q можна приписувати два значення 0 або 1, вільній предметній змінній y також два значення a або b. Таким чином, для даної формули можна задати всього $4\times 2\times 2 = 16$ різних приписувань значень змінних. Розглянемо конкретні два приписування

$$f_1\colon \begin{cases} Q \mapsto 1, \\ y \mapsto b, \\ P(x) \mapsto \lambda_2(x); \end{cases} \qquad f_2\colon \begin{cases} Q \mapsto 0, \\ y \mapsto a, \\ P(x) \mapsto \lambda_1(x), \end{cases}$$

і знайдемо істинностні значення даної формули для них. При f_1 формула має значення 1, оскільки

$$(\forall x)(\lambda_2(x) \longrightarrow 1) \vee (1 \wedge \lambda_2(b)) = (\lambda_2(a) \longrightarrow 1)(\lambda_2(b) \longrightarrow 1) \vee (1 \wedge 0) = 1 \cdot 1 \vee 0 = 1 \vee 0 = 1,$$

при f_2 формула приймає значення 0, оскільки

$$(\forall x)(\lambda_1(x) \longrightarrow 0) \vee (0 \wedge \lambda_1(a)) = (\lambda_1(a) \longrightarrow 0)(\lambda_1(b) \longrightarrow 0) \vee (0 \wedge 1) =$$
$$= (1 \longrightarrow 0)(1 \longrightarrow 0) = 0 \cdot 0 \vee 0 = 0 \vee 0 = 0.$$

4. Поняття рівносильності формул вводиться також і в логіці предикатів. А саме, дві формули A і B логіки предикатів називаються *рівносильними* (це позначається через $A \equiv B$), якщо *вони приймають однакові істинностні значення при кожному приписувані значень змінним над довільним полем.*

Теорема 21. *В логіці предикатів виконуються такі рівносильності:*

1. $\sim (\forall x)A(x) \equiv (\exists x) \sim A(x)$;
2. $\sim (\exists x)A(x) \equiv (\forall x) \sim A(x)$;
3. $(\forall x)(A(x) \wedge B) \equiv (\forall x)A(x) \wedge B$;
4. $(\exists x)(A(x) \wedge B) \equiv (\exists x)A(x) \wedge B$;
5. $(\forall x)(A(x) \vee B) \equiv (\forall x)A(x) \vee B$;
6. $(\exists x)(A(x) \vee B) \equiv (\exists x)A(x) \vee B$;
7. $(\forall x)(A(x) \longrightarrow B) \equiv (\exists x)A(x) \longrightarrow B$;
8. $(\exists x)(A(x) \longrightarrow B) \equiv (\forall x)A(x) \longrightarrow B$;
9. $(\forall x)(B \longrightarrow A(x)) \equiv B \longrightarrow (\forall x)A(x)$;
10. $(\exists x)(B \longrightarrow A(x)) \equiv B \longrightarrow (\exists x)A(x)$,

де предметна змінна x не входить вільно в формулу B.

Доведення. Доведемо, наприклад, властивість 9. Нехай $|(\forall x)(B \longrightarrow A(x))| = 0$, тобто знайдеться x_0 таке, що $|B \longrightarrow A(x_0)| = 0$. Звідси маємо $|B| = 1$ і $|A(x_0)| = 0$. Отже, $|(\forall x)A(x)| = 0$, тому $|B \longrightarrow (\forall x)A(x)| = 0$. Обернене доводиться аналогічно. Таким чином, формули $(\forall x)(B \longrightarrow A(x))$ і $B \longrightarrow (\forall x)A(x)$ одночасно приймають однакові значення, тобто вони рівносильні. □

Означення 6 *Формула $(Q_1x_1)\ldots(Q_nx_n)A$, де (Q_ix_i) є квантор загальності або існування, $x_i \neq x_j$ при $i \neq j$ і A не містить кванторів, називається випередженою нормальною формою.*

Очевидно, як випливає з теореми 21, кожна формула логіки предикатів може бути зведена до випередженої нормальної форми. Покажемо цю процедуру на конкретному прикладі.

Приклад. Знайти випереджену нормальну форму для формули:

$$(\forall x)A(x) \longrightarrow (\forall x)(B(x,y) \longrightarrow\sim (\forall z)C(y,z)).$$

Маємо

$$\begin{aligned}
&\equiv (\forall x)A(x) \longrightarrow (\forall x)(B(x,y) \longrightarrow\sim (\forall z)C(y,z)) \equiv\\
&\equiv (\exists x)(A(x) \longrightarrow (\forall x)(B(x,y) \longrightarrow\sim (\forall z)C(y,z))) \equiv\\
&\equiv (\exists x)(A(x) \longrightarrow (\forall u)(B(u,y) \longrightarrow\sim (\forall z)C(y,z))) \equiv\\
&\equiv (\exists x)(\forall u)(A(x) \longrightarrow (B(u,y) \longrightarrow\sim (\forall z)C(y,z))) \equiv\\
&\equiv (\exists x)(\forall u)(A(x) \longrightarrow (B(u,y) \longrightarrow (\exists z) \sim C(y,z))) \equiv\\
&\equiv (\exists x)(\forall u)(\exists z)(A(x) \longrightarrow (B(u,y) \longrightarrow\sim C(y,z))).
\end{aligned}$$

3.2 Загальнозначущість і виконуваність формул в логіці предикатів

Загальнозначущість і виконуваність формул логіки предикатів. Приклад формули, яка виконувана в нескінченному полі, але не виконувана у скінченному полі. Формулювання проблеми вирішення в логіці предикатів. Розв'язання проблеми вирішення для формул у випередженій нормальній формі, які містять квантори загальності, що передують кванторам існування. Проблема вирішення для формул з одномісними предикатами. Застосування мови логіки предикатів для запису математичних тверджень, логічний наслідок в логіці предикатів.

1. Формула логіки предикатів називається *загальнозначущою в даному полі*, якщо вона приймає значення "істина"при кожному приписуванні значень предикатним і вільним предметним змінним над цим полем. Якщо формула A загальнозначуща над довільним полем, то вона називається просто *загальнозначущою*. Цей факт позначається символом $\models A$. Доведемо, наприклад, наступна формула загальнозначуща:

$$\models (\forall x)P(x) \vee (\forall x)Q(x) \longrightarrow (\forall x)(P(x) \vee Q(x)). \tag{3.2.1}$$

Справді, припустимо, що формула (3.2.2) не є загальнозначущою. Це означає, що знайдеться поле і таке приписування f: $\begin{cases} P(x) \mapsto P^*(x), \\ Q(x) \mapsto Q^*(x) \end{cases}$ над цим полем,[12] що формула (3.2.2) приймає значення "хиба". Тоді, очевидно,

$$|(\forall x)P^*(x) \vee (\forall x)Q^*(x)| = 1 \quad \text{і} \quad |(\forall x)(P^*(x) \vee Q^*(x))| = 0.$$

З останньої рівності випливає, що в даному полі знайдеться такий елемент a, що $|P^*(a) \vee Q^*(a)| = 0$, звідки $|P^*(a)| = 0$ і $|Q^*(a)| = 0$. Таким чином, $|(\forall x)P^*(x)| = 0$ і $|(\forall x)Q^*(x)| = 0$, тому $|(\forall x)P^*(x) \vee (\forall x)Q^*(x)| = 0$. Ми отримали протиріччя. Отже, формула (3.2.2) загальнозначуща.

Далі, формула логіки предикатів називається *виконуваною в даному полі*, якщо над цим полем знайдеться таке приписування значень предикатним змінним і вільним предметним змінним, при якому формула приймає значення "істина". Формула називається *виконуваною*, якщо вона виконувана в деякому полі. Очевидно, що формула A загальнозначуща тоді і тільки тоді, коли формула $\sim A$ не є виконуваною, і формула A виконувана тоді і тільки тоді, коли $\sim A$ не є загальнозначущою.

Відмітимо без доведення такі дві теореми:

Теорема 22. *Якщо формула логіки предикатів загальнозначуща в деякому полі, то вона загальнозначуща в кожному полі тієї ж або меншої потужності.*

Теорема 23. *Якщо формула логіки предикатів виконувана в деякому полі, то вона виконувана в іншому полі тієї ж або більшої потужності.*

[12] Відмітимо, що нами позначено через $P^*(x), Q^*(x)$ логічні функції над даним полем, які приписуються предикатним змінним $P(x), Q(x)$.

Нехай A — формула, x — предметна змінна, що входить в A (в A, крім x, можуть бути й інші предметні змінні). Щоб показати залежність A від x, будемо писати $A(x)$. Будемо говорити, що формула $A(x)$ *вільна для* y, якщо в A відсутні вільні входження x, що входять в область дії кванторів $(\forall y)$ або $(\exists y)$. Наприклад, формула $P(x,y)\wedge(\forall y)Q(y)$ вільна для y, а формула $(x=1)\wedge(\exists y)(y\neq x)$ вже не є вільною для y.

Теорема 24. *Нехай $A(x)$ — формула, вільна для y. Тоді мають місце:*

1. $\models (\forall x)A(x)\longrightarrow A(y)$;

2. $\models A(y)\longrightarrow(\exists x)A(x)$.

Доведення. Справді, припустимо, що перша формула не загальнозначуща. Це означає, що в деякому полі знайдеться приписування, при якому $|(\forall x)A(x)\longrightarrow A(a)|=0$, де $y\mapsto a$ в даному приписуванні. Отже, $|(\forall x)A(x)|=1$ і $|A(a)|=0$. З першої умови випливає, що $|A(a)|=1$. Отримане протиріччя говорить про те, що припущення було невірним. Аналогічно доводиться друге твердження. □

Теорема 25. *Нехай x,y — різні предметні змінні, $P(x)$, $Q(x)$, $P(x,y)$ — формули, Q — довільна формула, яка не містить вільних входжень x, тоді*

1. $\models\sim(\forall x)P(x)\longleftrightarrow(\exists x)\sim P(x)$;

2. $\models\sim(\exists x)P(x)\longleftrightarrow(\forall x)\sim P(x)$;

3. $\models(\forall x)(P(x)\wedge Q(x))\longleftrightarrow(\forall x)P(x)\wedge(\forall x)Q(x)$;

4. $\models(\exists x)(P(x)\vee Q(x))\longleftrightarrow(\exists x)P(x)\vee(\exists x)Q(x)$;

5. $\models(\forall x)P(x)\vee(\forall x)Q(x)\longrightarrow(\forall x)(P(x)\vee Q(x))$;

6. $\models(\exists x)(P(x)\wedge Q(x))\longrightarrow(\exists x)P(x)\wedge(\exists x)Q(x)$;

7. $\models(\forall x)(P(x)\longrightarrow Q(x))\longrightarrow((\forall x)P(x)\longrightarrow(\forall x)Q(x))$;

8. $\models((\exists x)P(x)\longrightarrow(\exists x)Q(x))\longrightarrow(\exists x)(P(x)\longrightarrow Q(x))$;

9. $\models(\forall x)(P(x)\longleftrightarrow Q(x))\longrightarrow((\forall x)P(x)\longleftrightarrow(\forall x)Q(x))$;

10. $\models((\exists x)P(x)\longleftrightarrow(\exists x)Q(x))\longrightarrow(\exists x)(P(x)\longleftrightarrow Q(x))$;

11. $\models(\exists x)(P(x)\longrightarrow Q(x))\longleftrightarrow((\forall x)P(x)\longrightarrow(\exists x)Q(x))$;

12. $\models((\exists x)P(x)\longrightarrow(\forall x)Q(x))\longrightarrow(\forall x)(P(x)\longrightarrow Q(x))$;

13. $\models(\forall x)(P(x)\wedge Q)\longleftrightarrow((\forall x)P(x)\wedge Q)$;

14. $\models(\forall x)(P(x)\vee Q)\longleftrightarrow((\forall x)P(x)\vee Q)$;

15. $\models(\exists x)(P(x)\wedge Q)\longleftrightarrow((\exists x)P(x)\wedge Q)$;

16. $\models(\exists x)(P(x)\vee Q)\longleftrightarrow((\exists x)P(x)\vee Q)$;

17. $\models (\forall x)(P(x) \longrightarrow Q) \longleftrightarrow ((\exists x)P(x) \longrightarrow Q)$;

18. $\models (\exists x)(P(x) \longrightarrow Q) \longleftrightarrow ((\forall x)P(x) \longrightarrow Q)$;

19. $\models (\forall x)(P \longrightarrow Q(x)) \longleftrightarrow (P \longrightarrow (\forall x)Q(x))$;

20. $\models (\exists x)(P \longrightarrow Q(x)) \longleftrightarrow (P \longrightarrow (\exists x)Q(x))$;

21. $\models (\forall x)(\forall y)P(x,y) \longleftrightarrow (\forall y)(\forall x)P(x,y)$;

22. $\models (\exists x)(\exists y)P(x,y) \longleftrightarrow (\exists y)(\exists x)P(x,y)$;

23. $\models (\exists x)(\forall y)P(x,y) \longrightarrow (\forall y)(\exists x)P(x,y)$.

Доведення тверджень 1 – 23 проводиться за допомогою звичайних міркувань.

2. Щоб показати, що теорема 22 для здійсненності не має місця, наведемо приклад формули, яка виконувана в нескінченному полі, але не виконувана в жодному скінченному полі. Справді, розглянемо формулу:

$$(\forall x)(\forall y)(\forall z)(\exists u)\Big(\sim P(x,x) \wedge (P(x,y) \wedge P(y,z) \longrightarrow P(x,z)) \wedge P(x,u)\Big).$$

Припустимо, що ця формула виконується в деякому полі M. В такому випадку, існує двомісна логічна функція $\lambda(x,y)$ над цим полем, для якої дана формула істинна на M. Неважко бачити, що $\lambda(x,y)$ встановлює між елементами поля M відношення строгого порядку, тому що справедливі умови:

1. $\sim \lambda(x,x)$ для довільного $x \in M$ (антирефлексивність);

2. $\lambda(x,y) \wedge \lambda(y,z) \longrightarrow \lambda(x,z)$ для всіх $x,y,z \in M$ (транзитивність).

Якщо $\lambda(x,y)$, то будемо казати, що "x передує y". Візьмемо довільний елемент поля x_1, тоді серед елементів поля повинен знайтись елемент x_2, відмінний від x_1, такий, що "x_1 передує x_2". Точно так саме повинен знайтись елемент x_3 такий, що "x_2 передує x_3"і т.д. Отримуємо послідовність елементів: $x_1, x_2, \ldots, x_n, \ldots$. В силу умов 1 і 2 кожний елемент послідовності відмінний від всіх елементів з меншими індексами. Але це значить, що кожні два елемента послідовності різні, тому поле M нескінченне. Отже, ми довели, що *коли дана формула виконується в деякому полі, то воно нескінченне.*

Покажемо тепер, що існує поле, на якому дана формула виконується. Нехай $\mathbb{N}$ є множина натуральних чисел, а $P(x,y)$ означає, що $x < y$, де $x,y \in \mathbb{N}$. Тоді дана формула приймає вигляд:

$$(\forall x)(\forall y)(\forall z)(\exists u)\Big(x \not< x \wedge (x < y \wedge y < z \longrightarrow x < z) \wedge x < u\Big).$$

Легко бачити, що для натурального ряду цей вираз істинний.

3. *Проблема вирішення* в логіці предикатів формулюється так: вказати ефективний спосіб, за допомогою якого для кожної формули логіки предикатів можна було б визначити чи вона загальнозначуща, чи ні. Як показав відомий американський логік А. Черч, в загальному випадку ця проблема має негативний розв'язок, тобто не існує алгоритму,

за допомогою якого для довільної формули можна сказати, що вона загальнозначуща або не загальнозначуща. Однак, для деяких спеціальних класів формул ця задача має позитивний розв'язок. Далі ми розглянемо лише два з подібних класів.

а) Розв'язання проблеми вирішення для формул у випередженій нормальній формі, які містять квантори загальності, що передують кванторам існування.

Розглянемо формулу логіки предикатів

$$(\forall x_1)\dots(\forall x_m)(\exists y_1)\dots(\exists y_n)A(x_1,\dots x_m,y_1,\dots,y_n), \tag{3.2.2}$$

де $A(x_1,\dots x_m,y_1,\dots,y_n)$ є безкванторна формула.

Теорема 26. *Формула (3.2.2) є загальнозначущою тоді і тільки тоді, коли вона загальнозначуща в полі з m елементів.*

Доведення. Нехай формула (3.2.2) є загальнозначущою, тому вона загальнозначуща в довільному полі, в тому числі, в полі з m елементів.

Навпаки, нехай (3.2.2) загальнозначуща в m-елементному полі. Виберемо довільне поле M потужності більше, ніж m. Розглянемо формулу

$$(\exists y_1)\dots(\exists y_n)A(x_1,\dots x_m,y_1,\dots,y_n), \tag{3.2.3}$$

де $x_1,\dots,x_m$ вже є вільними предметними змінними. Очевидно, формули (3.2.2) і (3.2.3) в полі M одночасно загальнозначущі або ні. Розглянемо тепер довільне приписування змінним формули (3.2.3) над полем M, і нехай при цьому $x_i \mapsto a_i$, $i=1,\dots,m$, $a_i \in M$. В результаті приписування формула (3.2.3) буде мати вид:

$$(\exists y_1)\dots(\exists y_n)A(a_1,\dots a_m,y_1,\dots,y_n). \tag{3.2.4}$$

Позначимо через M_0 множину елементів $\{a_1,\dots,a_m\}$. За умовою теореми формула (3.2.3) загальнозначуща в M_0, тому і формула (3.2.4) загальнозначуща в M_0. Очевидно, що (3.2.4) в полі M рівносильна формулі

$$\bigvee_{(b_1,\dots,b_n)\in M^n} A(a_1,\dots,a_m,b_1,\dots,b_n), \tag{3.2.5}$$

яку ми можемо переписати у вигляді:

$$\bigvee_{(b_1,\dots,b_n)\in M_0^n} A(a_1,\dots,a_m,b_1,\dots,b_n) \vee \bigvee_{(b_1,\dots,b_n)\in M^n\setminus M_0^n} A(a_1,\dots,a_m,b_1,\dots,b_n). \tag{3.2.6}$$

Але оскільки (3.2.4) загальнозначуща в M_0, то, очевидно, формула

$$\bigvee_{(b_1,\dots,b_n)\in M_0^n} A(a_1,\dots,a_m,b_1,\dots,b_n)$$

має значення "істина", тому вся формула (3.2.6) істинна. Ми показали, що формула (3.2.2) приймає значення "істина" при довільному приписуванні над полем M. Отже, формула (3.2.2) загальнозначуща. □

б) Проблема вирішення для формул з одномісними предикатами.

Теорема 27. *Формула логіки предикатів, яка містить n різних одномісних предикатних змінних, загальнозначуща тоді і тільки тоді, коли вона загальнозначуща в полі, що містить 2^n елементів.*

Доведення. Необхідність теореми очевидна, тому зупинимось лише на доведенні достатності. Отже, розглянемо формулу логіки предикатів у випередженій нормальній формі

$$(Qx_1)(Qx_2)\dots(Qx_m)A(P_1,\dots,P_n,x_1,\dots,x_m), \tag{3.2.7}$$

де (Qx_i) є або $(\forall x_i)$, або $(\exists x_i)$, $i=1,\dots,m$, а формула $A(P_1,\dots,P_n,x_1,\dots,x_m)$ кванторів не містить, де $P_1,\dots,P_n$— одномісні предикатні змінні, а $x_1,\dots,x_n$— предметні змінні, що входять в них. Оскільки кожна предметна змінна x_i $(i=1,\dots,n)$ входить хоч б в один з предикатів $P_1,\dots,P_n$, то очевидно, що $m\leqslant n$, тому ми замість $A(P_1,\dots,P_n,x_1,\dots,x_m)$ інколи будемо писати $A(P_1(x_{i_1}),\dots,P_n(x_{i_n}))$ де $i_1,\dots,i_n\in\{1,\dots,m\}$.

Припустимо, що формула (3.2.7) загальнозначуща в полі з 2^n елементів, і покажемо, що вона загальнозначуща в довільному полі з більшим, ніж 2^n, числом елементів. Нехай D є довільне поле таке, що $|D|>2^n$. Розглянемо над цим полем довільне приписування f формули (3.2.7). Припустимо, що при цьому f: $x_k\mapsto a_k$, $k=1,\dots,m$, f: $P_l\mapsto\lambda_l$, $l=1,\dots,n$. Визначимо на D відношення еквівалентності ε таким чином:

$$d_1\equiv d_2(\varepsilon)\overset{df}{\longleftrightarrow}\lambda_1(d_1)=\lambda_1(d_2)\wedge\lambda_2(d_1)=\lambda_2(d_2)\wedge\ldots\wedge\lambda_n(d_1)=\lambda_n(d_2).$$

Нехай $\bar\alpha=(\alpha_1,\dots,\alpha_n)$ є деякий двійковий набір. Поставимо йому у відповідність підмножину $H_{\bar\alpha}$, яка визначається умовою:

$$d\in H_{\bar\alpha}\overset{df}{\longleftrightarrow}\lambda_1(d)=\alpha_1\wedge\lambda_2(d)=\alpha_2\wedge\ldots\wedge\lambda_n(d)=\alpha_n,$$

де $d\in D$. Очевидно, що $d\in H_{(\lambda_1(d),\dots,\lambda_n(d))}$ для кожного $d\in D$. Далі, якщо $\bar\alpha\neq\bar\beta$, то $H_{\bar\alpha}\cap H_{\bar\beta}=\varnothing$. Отже, сім'я підмножин $(H_{\bar\alpha_i})_{i=1,\dots,2^n}$ утворює розбиття множини D. Причому, якщо $d_1,d_2\in H_{\bar\alpha}$, то, очевидно, $d_1\equiv d_2(\varepsilon)$. Таким чином, $D/\varepsilon=(H_{\bar\alpha_i})_{i=1,\dots,2^n}$. Отже, поле D/ε має 2^n елементів. Через $[\,d\,]$ позначимо ε-клас, який містить елемент d. Розглянемо над полем D/ε приписування f^0, яке визначається таким чином:

$$f^0\colon\begin{cases}x_k\mapsto[a_k], & k=1,\dots,m;\\ P_l\mapsto\lambda^0_l, & l=1,\dots,n,\end{cases}$$

де $\lambda^0_l([\,d\,])\overset{df}{=}\lambda_l(d)$. Покажемо тепер, що формула $A(P_1,\dots,P_n,x_1,\dots,x_m)$ в приписуваннях f і f^0 приймають однакові істинностні значення. Справді,

$$\begin{aligned}&A(\lambda^0_1(x_{i_1}),\dots,\lambda^0_n(x_{i_n}),[\,a_1\,],\dots,[\,a_m\,])\equiv\\ \equiv\;&A(\lambda^0_1(a_{i_1}),\dots,\lambda^0_n(a_{i_n}))=A(\lambda_1(a_{i_1}),\dots,\lambda_n(a_{i_n}))\equiv\\ \equiv\;&A(\lambda_1(x_{i_1}),\dots,\lambda_n(x_{i_n}),a_1,\dots,a_m).\end{aligned}$$

Покажемо тепер, що формула

$$(Qx_m)A(P_1,\ldots,P_n,x_1,\ldots,x_{m-1},x_m), \tag{3.2.8}$$

де (Qx_m) є $(\forall x_m)$ або $(\exists x_m)$, при приписуваннях f і f^0 також приймає однакові істинностні значення. Припустимо конкретно, що (Qx_m) є $(\forall x_m)$, тоді формула (3.2.8) набуває вигляду:

$$(\forall x_m)A(P_1,\ldots,P_n,x_1,\ldots,x_{m-1},x_m). \tag{3.2.9}$$

Не втрачаючи загальності, припустимо далі, що $P_n = P_n(x_m)$. Виберемо потім по одному представнику з кожного ε-класу і позначимо їх відповідно через $d_1, d_2, \ldots, d_{2^n}$. Таким чином, будемо мати $D/\varepsilon = \{[d_1],[d_2],\ldots,[d_{2^n}]\}$. Далі маємо:

$$\begin{aligned}
&(\forall x_m \in D/\varepsilon)A(\lambda_1^0(x_{i_1}),\ldots,\lambda_{n-1}^0(x_{i_{n-1}}),\lambda_n^0(x_m),[a_1],\ldots,[a_{m-1}],x_m) \equiv\\
\equiv\ &(\forall x_m \in D/\varepsilon)A(\lambda_1^0([a_{i_1}]),\ldots,\lambda_{n-1}^0([a_{i_{n-1}}]),\lambda_n^0(x_m)) \equiv\\
\equiv\ &\bigwedge_{k=1}^{2^n} A(\lambda_1^0([a_{i_1}]),\ldots,\lambda_{n-1}^0([a_{i_{n-1}}]),\lambda_n^0([d_k])) \equiv\\
\equiv\ &\bigwedge_{k=1}^{2^n}\bigwedge_{d\in[d_k]} A(\lambda_1^0([a_{i_1}]),\ldots,\lambda_{n-1}^0([a_{i_{n-1}}]),\lambda_n^0([d])) \equiv\\
\equiv\ &\bigwedge_{k=1}^{2^n}\bigwedge_{d\in[d_k]} A(\lambda_1(a_{i_1}),\ldots,\lambda_{n-1}(a_{i_{n-1}}),\lambda_n(d)) \equiv\\
\equiv\ &\bigwedge_{d\in D} A(\lambda_1(a_{i_1}),\ldots,\lambda_{n-1}(a_{i_{n-1}}),\lambda_n(d)) \equiv\\
\equiv\ &(\forall x_m \in D)A(\lambda_1(a_{i_1}),\ldots,\lambda_{n-1}(a_{i_{n-1}}),\lambda_n(x_m)) \equiv\\
\equiv\ &(\forall x_m \in D)A(\lambda_1(x_{i_1}),\ldots,\lambda_{n-1}(x_{i_{n-1}}),\lambda_n(x_m),a_1,\ldots,a_{m-1},x_m).
\end{aligned}$$

Отже, формула (3.2.9) при приписуваннях f і f^0 приймають однакові істинностні значення.

Якщо ж (Qx_m) є $(\exists x_m)$, то аналогічне твердження також справедливе, що перевіряється подібними міркуваннями. Міркуючи таким же чином і далі, ми в результаті покажемо, що формула (3.2.7) при приписуваннях f і f^0 приймає однакові істинностні значення. Враховуючи тепер, зо (3.2.7) загальнозначуща за умовою теореми в полі D/ε, оскільки воно містить 2^n елементів, робимо висновок, що формула (3.2.7) при приписуванні f^0 приймає значення "істина", тому вона також і при приписуванні f набуває значення "істина". В силу довільності вибору приписування f над полем D, робимо висновок, що формула (3.2.7) приймає значення "істина"при довільному приписуванні над цим же полем, тобто вона буде загальнозначущою в полі D. Оскільки поле D вибиралось також довільно, лише б $|D| > 2^n$, томі (3.2.7) загальнозначуща у всякому полі з числом елементів більшим, ніж 2^n, а тому формула (3.2.7) загальнозначуща, що і треба було довести. □

Приклад. Довести, що формула

$$(\forall x)P(x) \vee (\forall x)Q(x) \longrightarrow (\forall x)(P(x) \vee Q(x)) \tag{3.2.10}$$

загальнозначуща.

Справді, формула (3.2.10) буде загальнозначущою тоді і тільки тоді, коли вона, згідно теореми 27, буде загальнозначуща в полі з $2^2 = 4$ (чотирьох) елементів, оскільки вона містить дві предикатні одномісні змінні. Розглянемо поле $D = \{a, b, c, d\}$ і запишемо формулу (3.2.10) над ним у вигляді:

$$P(a)P(b)P(c)P(d) \vee Q(a)Q(b)Q(c)Q(d) \longrightarrow$$
$$\longrightarrow (P(a) \vee Q(a))(P(b) \vee Q(b))(P(c) \vee Q(c))(P(d) \vee Q(d)).$$

Припустимо, що P, Q є логічні функції при довільному приписуванні над D і записана формула хибна. Тоді

$$|P(a)P(b)P(c)P(d) \vee Q(a)Q(b)Q(c)Q(d)| = 1,$$
$$|(P(a) \vee Q(a))(P(b) \vee Q(b))(P(c) \vee Q(c))(P(d) \vee Q(d))| = 0.$$

Нехай $P(a) \vee Q(a)$ є "хиба", тоді $|P(a)| = 0$ і $|Q(a)| = 0$. Тому $|P(a)P(b)P(c)P(d)| = 0$ і $|Q(a)Q(b)Q(c)Q(d)| = 0$, отже, $|P(a)P(b)P(c)P(d) \vee Q(a)Q(b)Q(c)Q(d)| = 0$. Ми отримали протиріччя, яке говорить про те, що наше припущення було невірним. Таким чином, формула (3.2.10) загальнозначуща в полі D, тому згідно теореми вона загальнозначуща.

4. Покажемо на прикладах як логіка предикатів використовується для записування математичних означень та тверджень.

Приклад 1. Нехай змінні x, y, z приймають значення з множини $\mathbb{R}$ дійсних чисел. Записати символічно: "Для кожного дійсного числа x існує таке y, що для кожного z, якщо сума z і 1 менше y, то сума x і 2 менше 4".

$$(\forall x)(\exists y)(\forall z)(z + 1 < y \longrightarrow x + 2 < 4).$$

Приклад 2. Означення границі послідовності:

$$a = \lim_{n\to\infty} x_n \overset{df}{\longleftrightarrow} (\forall \varepsilon > 0)(\exists n_0 \in \mathbb{N})(\forall n \in \mathbb{N})(n > n_0 \longrightarrow |x_n - a| < \varepsilon).$$

Приклад 3. Означення границі функції в точці:

$$A = \lim_{x\to a} f(x) \overset{df}{\longleftrightarrow} (\forall \varepsilon > 0)(\exists \delta > 0)(\forall x)(|x - a| < \delta \longrightarrow |f(x) - A| < \varepsilon).$$

В логіці предикатів поняття логічного наслідку вводиться таким же самими чином, як це було зроблено в свій час в логіці висловлень, а саме, *формула B логіки предикатів вважається логічним наслідком формул логіки предикатів $A_1, A_2, \ldots, A_n$, які називаються посилками, тоді і тільки тоді, коли формула $A_1 \wedge A_2 \wedge \ldots \wedge A_n \longrightarrow B$ є загальнозначущою*, тобто

$A_1, A_2, \ldots, A_n \models B$ тоді і тільки тоді, коли $\models A_1 \wedge A_2 \wedge \ldots \wedge A_n \longrightarrow B$.

Приклад 4. Деякі хіміки — велосипедисти. Жоден філософ не є хіміком. Отже, деякі велосипедисти не є філософами. Чи буде правильним таке міркування з логічної точки зору?

Розв'язування. Введемо такі позначення: $X(x)$ — "x є хімік", $B(x)$ — "x є велосипедист", $\Phi(x)$ — "x є філософ". Тоді умова задачі в символічній формі запишеться так:

$$(\exists x)(X(x) \wedge B(x)), (\forall x)(\Phi(x) \longrightarrow\sim X(x)) \models (\exists x)(B(x)\wedge \sim \Phi(x))$$

або таким чином

$$\begin{array}{l} 1.\ (\exists x)(X(x) \wedge B(x)), \\ \underline{2.\ (\forall x)(\Phi(x) \longrightarrow\sim X(x))} \\ \therefore\ (\exists x)(B(x)\wedge \sim \Phi(x)) \end{array}$$

Розглянемо формулу логіки предикатів

$$(\exists x)(X(x) \wedge B(x)) \wedge (\forall x)(\Phi(x) \longrightarrow\sim X(x)) \longrightarrow (\exists x)(B(x)\wedge \sim \Phi(x)) \tag{3.2.11}$$

і перевіримо чи буде вона загальнозначущою. Припустимо, що ця формула не є загальнозначущою. Це означає, що існує таке поле D, а на ньому знайдеться таке приписування φ: $X(x) \mapsto X^*(x), B(x) \mapsto B^*(x), \Phi(x) \mapsto \Phi^*(x)$, при якому формула (3.2.11) є хибною, тобто

$$|(\exists x)(X^*(x) \wedge B^*(x)) \wedge (\forall x)(\Phi^*(x) \longrightarrow\sim X^*(x)) \longrightarrow (\exists x)(B^*(x)\wedge \sim \Phi^*(x))| = 0. \tag{3.2.12}$$

Умова (3.2.12) означає, що має місце наступна система умов:

$$|(\exists x)(X^*(x) \wedge B^*(x))| = 1, \tag{3.2.13}$$

$$|(\forall x)(\Phi^*(x) \longrightarrow\sim X^*(x))| = 1, \tag{3.2.14}$$

$$|(\exists x)(B^*(x)\wedge \sim \Phi^*(x))| = 0. \tag{3.2.15}$$

Рівність (3.2.13) означає, що існує таке $x_0 \in D$, при якому $|X^*(x_0) \wedge B^*(x_0)| = 1$, тобто $|X^*(x_0)| = 1$ і $|B^*(x_0))| = 1$. Умова (3.2.14) означає, що для кожного $x \in D$ має місце $|\Phi^*(x) \longrightarrow\sim X^*(x)| = 1$, звідки випливає, що $|\Phi^*(x_0) \longrightarrow\sim X^*(x_0)| = 1$. Оскільки $| \sim X^*(x_0)| = 0$, то очевидно $|\Phi^*(x_0)| = 0$, що випливає з означення імплікації. Рівність (3.2.15) означає, що для всіх $x \in D$ виконується $|B^*(x)\wedge \sim \Phi^*(x)| = 0$, тому $|B^*(x_0)\wedge \sim \Phi^*(x_0)| = 0$. Оскільки $| \sim \Phi^*(x_0)| = 1$, очевидно, $|B^*(x_0))| = 0$. Отримане протиріччя говорить про те, що наше припущення є невірним. Отже, формула (3.2.11) є загальнозначущою, що означає правильність міркування з логічної точки зору. □

4 Математичні теорії першого порядку

4.1 Означення теорії першого порядку. Числення предикатів

Мова першого порядку. Терми і формули. Логічні та спеціальні аксіоми. Правила виведення. Приклади математичних теорій. Доведення в теорії першого порядку. Теорема дедукції.

1. Аксіоматичний метод, який, мабуть, був лишньою розкішшю при вивченні логіки висловлень, є необхідним при вивченні логіки предикатів, що пояснюється в першу чергу відсутністю алгоритму, за допомогою якого можна було б розпізнавати загальнозначущі формули. Таким чином, ми приходимо до розглядання *теорій першого порядку* (або, інакше, *елементарних теорій*). Відмітимо, що в елементарних теоріях квантори можуть навішуватись тільки на предметні змінні, і, ні в якому разі, на предикатні змінні, а також не дозволяються предикати, які мають в якості можливих значень своїх аргументів інші предикати та функції.

Символами кожної *теорії* $\mathbf{K}$ *першого порядку* служать: *логічні зв'язки* $\sim$, $\longrightarrow$; *знаки пунктуації* $(\ ,\ )$, $,$; *непорожня*, скінченна бо зчисленна, множина *предикатних змінних* A_j^n $(n \geqslant 0,\ j \geqslant 1)$; скінченна (можливо, і порожня) або зчисленна множина *функціональних змінних* f_j^n $(n, j \geqslant 1)$; і, нарешті, скінченна (можливо, порожня) або зчисленна множина *предметних констант* a_i $(i \geqslant 1)$; $\forall$ — квантор загальності. Верхній індекс предикатної або функціональної змінної вказує число аргументів, а нижній індекс служить щоб розрізняти літери з одним і тим же числом аргументів.

Функціональні змінні, що застосовуються до предметних змінних і констант, породжують *терми*. А саме,

(a) кожна предметна змінна або предметна константа є терм;

(b) якщо f_i^n — функціональна змінна, а $t_1, \ldots, t_n$— терми, то $f_i^n(t_1, \ldots, t_n)$ є терм;

(c) вираз є термом тільки у тому випадку, коли він випливає з правил (a) і (b).

Предикатні змінні, застосовані до термів, породжують *елементарні формули*, або точніше: якщо A_i^n— предикатна змінна, а $t_1, \ldots, t_n$— терми, то $A_i^n(t_1, \ldots, t_n)$— елементарна формула.

Формули визначаються таким чином:

(a) кожна елементарна формула є формула;

(b) якщо A і B— формули і y— предметна змінна, то кожний з виразів $(\sim A)$, $(A \longrightarrow B)$ і $((\forall y)A)$ є формула;

(c) вираз є формулою тільки в тому разі, коли він випливає з правил (a) і (b).

Звісно, у випадку кожної конкретної теорії $\mathbf{K}$ в побудові термів і формул беруть участь тільки ті символи, які належать теорії $\mathbf{K}$.

Терм t називається *вільним для предметної змінної* x_i у формулі A, якщо жодне вільне входження x_i в A не знаходиться в області дії жодного квантора $\forall x_j$, де x_j— предметна змінна, що входить в t. Наприклад, терм $f_1^2(x_1, x_3)$ вільний для x_1 в $(\forall x_2)A_1^2(x_1, x_2) \longrightarrow A_1^1(x_1)$, але не вільний для x_1 в $(\exists x_3)(\forall x_2)A_1^2(x_1, x_2) \longrightarrow A_1^1(x_1)$.

Аксіоми теорії $\mathbf{K}$ розбиваються на два класи: логічні аксіоми і власні (або нелогічні) аксіоми.

Логічні аксіоми: які б не були формули A, B і C теорії $\mathbf{K}$, наступні формули є логічними аксіомами теорії $\mathbf{K}$:

(1) $A \longrightarrow (B \longrightarrow A)$;

(2) $(A \longrightarrow (B \longrightarrow C)) \longrightarrow ((A \longrightarrow B) \longrightarrow (A \longrightarrow C))$;

(3) $(\sim B \longrightarrow \sim A) \longrightarrow ((\sim B \longrightarrow A) \longrightarrow B)$

(4) $(\forall x_i)A(x_i) \longrightarrow A(t)$, де $A(x_i)$ є формула теорії $\mathbf{K}$ і t— терм теорії $\mathbf{K}$, вільний для x_i в $A(x_i)$.

(5) $(\forall x_i)(A \longrightarrow B) \longrightarrow (A \longrightarrow (\forall x_i)B)$, якщо формула A не містить вільних входжень предметної змінної x_i.

Власні аксіоми: вони не можуть бути сформульовані в загальному випадку, оскільки змінюються від теорії до теорії. Теорія першого порядку, яка не містить власних аксіом, називається *численням предикатів першого порядку*.

Правилами виведення в кожній теорії першого порядку є:

(a) *Modus ponens*: з A і $A \longrightarrow B$ випливає B. (Скорочене позначення MP)

(b) *Правило узагальнення*: з A випливає $(\forall x_i)A$. (Скорочене позначення Gen)

2. Приклад 1. *Теорія часткового порядку.* Нехай $\mathbf{K}$ містить одну предикатну змінну A_1^2 і не містить функціональних змінних та предметних констант. Замість $A_1^2(x_1, x_2)$ і $\sim A_1^2(x_1, x_2)$ будемо відповідно писати $x_1 < x_2$ і $x_1 \not< x_2$. Нехай, нарешті, $\mathbf{K}$ містить дві власні аксіоми:

(a) $(\forall x_1)(x_1 \not< x_1)$ (іррефлексивність);

(b) $(\forall x_1)(\forall x_2)(\forall x_3)(x_1 < x_2 \wedge x_2 < x_3 \longrightarrow x_1 < x_3)$ (транзитивність).

Відмітимо, що $\wedge, \vee$ і $\longleftrightarrow$ визначаються через $\sim, \longrightarrow$ точно так, як в численні висловлень, і $(\exists x)A$ означає $\sim (\forall x) \sim A$.

Приклад 2. *Теорія груп.* Нехай $\mathbf{K}$ має одну предикатну змінну A_1^2, одну функціональну змінну f_1^2 і одну предметну константу a_1. Замість $A_1^2(t, s)$, $f_1^2(t, s)$, a_1 будемо писати відповідно $t = s$, ts, e. Власними аксіомами теорії $\mathbf{K}$ є формули:

(a) $(\forall x_1)(\forall x_2)(\forall x_3)(x_1(x_2x_3) = (x_1x_2)x_3)$; (асоціативність)

(b) $(\forall x_1)(ex_1 = x_1)$; ($e$ — ліва одиниця)

(c) $(\forall x_1)(\exists x_2)(x_2x_1 = e)$; (існування лівого оберненого елемента)

(d) $(\forall x_1)(x_1 = x_1)$; (рефлексивність рівності)

(e) $(\forall x_1)(\forall x_2)(x_1 = x_2 \longrightarrow x_2 = x_1)$; (симетричність рівності)

(f) $(\forall x_1)(\forall x_2)(\forall x_3)(x_1 = x_2 \longrightarrow (x_2 = x_3 \longrightarrow x_1 = x_3))$;(транзитивність рівності)

(g) $(\forall x_1)(\forall x_2)(\forall x_3)(x_2 = x_3 \longrightarrow (x_1x_2 = x_1x_3 \wedge x_2x_1 = x_3x_1))$. (підстановочність рівності)

3. Відмітимо, перш за все, що поняття доведення, вивідності з гіпотез залишаються такими ж для теорій першого порядку, якими ми їх давали для довільних формальних теорій.

Теорема дедукції, яка доведена в численні висловлень, не може бути чисто механічно без відповідної модифікації перенесена в теорію першого порядку $\mathbf{K}$. Наприклад, згідно вивідності з гіпотез має місце в теорії $\mathbf{K}$ твердження $A \vdash_{\mathbf{K}} (\forall x_1)A$ для довільної формули A цієї теорії. Однак не завжди $\vdash_{\mathbf{K}} A \longrightarrow (\forall x_1)A$. Справді, розглянемо поле $D = \{a, b\}$ і нехай A є $A^1_1(x_1)$. Розглянемо над полем D приписування $f\colon A^1_1 \mapsto \lambda$, де $\lambda(a) = 1$ і $\lambda(b) = 0$, $f\colon x_1 \mapsto a$. При такому приписуванні маємо $|\lambda(a)| = 1$, $|(\forall x_1)\lambda(x_1)| = 0$, тому $|\lambda(a) \longrightarrow (\forall x_1)\lambda(x_1)| = 0$. Ми показали, що формула $A^1_1(x_1) \longrightarrow (\forall x_1)\lambda(x_1)$ не є загальнозначущою. Однак деяка послаблена форма теореми дедукції у випадку теорій першого порядку може бути доведена.

Нехай A — деяка формула, що належить множині гіпотез Γ, і нехай $B_1, \ldots, B_n$ — який-небудь вивід з Γ, наділений обґрунтуванням кожного кроку у ньому. Ми будемо казати, що B_i *залежить від* A *у цьому виведенні*, якщо

(1) B_i є A і обґрунтуванням B_i слугує належність B_i до Γ; або

(2) B_i обґрунтоване як безпосередній наслідок згідно MP або Gen деяких попередніх у цьому виведенні формул, з яких принаймні одна залежить від A.

Лема 12. *Якщо формула A теорії першого порядку $\mathbf{K}$ є частинний випадок тавтології, то A є теорема $\mathbf{K}$ і може бути виведена із застосуванням одних тільки схем аксіом (1) – (3) і правила modus ponens.*

Доведення. Нехай A отримана з деякої тавтології W за допомогою підстановок. Тоді, очевидно, існує вивід W в численні висловлень $\mathbf{L}$. Зробимо тепер всюди в цьому виведенні підстановки за правилом: (a) якщо яка-небудь логічна змінна входить в W, то на місця всіх її входжень в кожну формулу виведення підставляємо ту формулу теорії $\mathbf{K}$, яка підставлялася в W на місця входжень тієї ж літери при побудові A; (b) якщо дана логічна змінна не входить в W, то на місця всіх її входжень у формули виведення підставляємо довільну формулу теорії $\mathbf{K}$. Отримана таким чином послідовність формул і буде виведенням формули A в $\mathbf{K}$, причому виведенням, який використовує лише схеми аксіом (1) – (3) і MP. □

Лема 13. *Якщо B не залежить від A у виведенні $\Gamma, A \vdash B$, то $\Gamma \vdash B$.*

Доведення. Нехай $B_1, \ldots, B_n(= B)$ — виведення B з Γ і A, в якому B не залежить від A. Візьмемо за індуктивне припущення, що твердження, яке доводиться, справедливе для всіх виведень, довжина яких менше n. Якщо B належить Γ або є аксіома, то $\Gamma \vdash B$. Якщо B є безпосереднім наслідком яких-небудь попередніх формул, то, оскільки B не залежить від A, то не залежить від A жодна з цих формул. Отже, згідно індуктивного припущення, з Γ виводяться ці формули, а разом з ними і B. □

Теорема 28 (Метатеорема дедукції). *Нехай* $\Gamma, A \vdash B$*, і при цьому нехай існує таке виведення* B *з* $\{\Gamma, A\}$*, в якому при жодному застосуванні правила узагальнення до формул, що залежать у цьому виведенні від* A*, не зв'язується квантором жодна вільна змінна формули* A*. Тоді* $\Gamma \vdash A \longrightarrow B$.

Доведення. Нехай $B_1, \ldots, B_n (= B)$ — виведення B з $\{\Gamma, A\}$, що задовольняє умову теореми. Доведемо індукцією, що $\Gamma \vdash A \longrightarrow B_i$ для кожного $i \leqslant n$. Якщо B_i є аксіома або належить Γ, то $\Gamma \vdash A \longrightarrow B_i$, оскільки $A \longrightarrow B_i$ випливає згідно MP з B_i і аксіоми $B_i \longrightarrow (A \longrightarrow B_i)$. Якщо B_i співпадає з A, то $\Gamma \vdash A \longrightarrow B_i$ в силу того, що $\vdash A \longrightarrow A$ (лема 9). Якщо існують $j, k < i$ такі, що B_k є $B_j \longrightarrow B_i$, то, згідно індуктивному припущенню, $\Gamma \vdash A \longrightarrow B_j$ і $\Gamma \vdash A \longrightarrow (B_j \longrightarrow B_i)$. Отже, за схемою аксіом (2) і MP маємо $\Gamma \vdash A \longrightarrow B_i$. Нехай, нарешті, існує $j < i$ таке, що B_i є $(\forall x_k) B_j$. За припущенням $\Gamma \vdash A \longrightarrow B_j$, і або B_j не залежить від A, або x_k не є вільною предметною змінною формули A. Якщо B_j не залежить від A, то в силу леми 13 маємо $\Gamma \vdash B_j$, і тоді застосовуючи правило Gen, отримуємо $\Gamma \vdash (\forall x_k) B_j$, тобто $\Gamma \vdash B_i$. За схемою аксіом (1) маємо $\vdash B_i \longrightarrow (A \longrightarrow B_i)$, звідки за MP отримуємо $\Gamma \vdash A \longrightarrow B_i$.

Якщо x_k не є вільною предметною змінною формули A, то за схемою аксіом (5) маємо $\vdash (\forall x_k)(A \longrightarrow B_j) \longrightarrow (A \longrightarrow (\forall x_k) B_j)$. Оскільки $\Gamma \vdash B_j$, то за правилом Gen отримуємо $\Gamma \vdash (\forall x_k)(A \longrightarrow B_j)$. Тому за допомогою правила modus ponens ми виводимо $\Gamma \vdash A \longrightarrow (\forall x_k) B_j$, тобто $\Gamma \vdash A \longrightarrow B_i$. Цим і завершується індукція. Покладаючи тепер $i = n$ ми отримуємо твердження теореми $\Gamma \vdash A \longrightarrow B$. □

Наслідок 12. *Якщо* $\Gamma, A \vdash B$ *і існує виведення, побудоване без використання правила узагальнення до вільних предметних змінних формули* A*, то* $\Gamma \vdash A \longrightarrow B$.

Наслідок 13. *Якщо формула* A *замкнена*[13] *і* $\Gamma, A \vdash B$*, то* $\Gamma \vdash A \longrightarrow B$.

[13] Формула логіки предикатів називається *замкненою*, якщо вона немає вільних предметних змінних.

4.2 Несуперечність і повнота числення предикатів

Теорема про несуперечність числення предикатів першого порядку. Інтерпретації. Виконуваність та істинність. Моделі. Ізоморфізм моделей і категоричність. Повнота числення предикатів першого порядку.

1. Має місце теорема про несуперечливість числення предикатів.

Теорема 29. *Числення предикатів першого порядку є несуперечлива теорія.*

Доведення. Для довільної формули A позначимо через $h(A)$ вираз, який отримується в результаті витирання в A всіх кванторів і термів разом з відповідними дужками й комами. По суті $h(A)$ є формулою числення висловлень, в якій роль логічних змінних грають символи A_j^k. Очевидно, що $h(\sim A) = \sim h(A)$ і $h(A \longrightarrow B) = h(A) \longrightarrow h(B)$. Для кодної аксіоми A, яка отримується за якою-небудь схемою аксіом (1) – (5), $h(A)$ є тавтологією. Це очевидно для (1) – (3). Кожний частинний випадок $(\forall x_i)A(x_i) \longrightarrow A(t)$ схеми (4) перетворюється операцією h в тавтологію виду $B \longrightarrow B$, а кожний частинний випадок $(\forall x_i)(A \longrightarrow B) \longrightarrow (A \longrightarrow (\forall x_i)B)$ схеми (5) перетворюється в тавтологію виду $(D \longrightarrow E) \longrightarrow (D \longrightarrow E)$. Нарешті, якщо $h(A)$ і $h(A \longrightarrow B)$ — тавтології, то і $h(B)$ — тавтологія, і якщо $h(A)$ — тавтологія, то і $h((\forall x_i)A)$ — тавтологія, оскільки результати застосування операції h до A і $(\forall x_i)A$ співпадають. Отже, якщо A є теорема в численні предикатів, то $h(A)$ є тавтологія. Якби існувала формула B числення предикатів така, що $\vdash B$ і $\vdash \sim B$, то обидва вирази $h(B)$ і $h(\sim B)$ були б тавтологіями, що неможливо. Таким чином, числення предикатів першого порядку несуперечливе. □

2. Формули теорії першого порядку мають зміст, коли існує яка-небудь інтерпретація символів, що входять у неї. Під *інтерпретацією* ми будемо розуміти кожну систему, яка складається з непорожньої множини D, що називається *областю інтерпретації* (або *полем*), і деякої відповідності, яка кожній предикатній змінній A_j^n приписує деяке n-місне відношення в D, кожній функціональній змінній f_j^n — деяку n-арну операцію в D і кожній предметній сталій a_i — деякий елемент з D. При заданій інтерпретації предметні змінні розуміються пробігаючими область D цієї інтепретації, а зв'язкам і кванторам надається їх звичайний зміст.

При даній інтерпретації кожна формула без вільних предметних змінних (або, інакше *замкнена формула*) являє собою висловлення, яке істинне або хибне, а кожна формула з вільними предметними змінними визначає деяке відношення на області інтепретації; це відношення може бути виконане (істинне) для одних значень змінних із області інтерпретації і не виконане (хибне) для інших.

Нехай задана деяка інтерпретація з областю D, і нехай Σ є множина всіх зчисленних послідовностей елементів з D. Будемо казати, що формула A *виконується на послідовності* $s = (b_1, b_2, \ldots)$ *з* Σ *при даній інтерпретації*, якщо для кожного $i = 1, 2, \ldots$ підстановка b_i на місця всіх вільних входжень x_i в A призводить до істинного в даній інтерпретації твердження. Формула A називається *істинною в даній інтерпретації* тоді і тільки тоді, коли вона виконується на кожній послідовності з Σ. Формула A називається *хибною в даній інтерпретації*, якщо вона не виконується на жодній послідовності з Σ.

Дана інтерпретація називається *моделлю* для даної множини формул Γ, якщо кожна формула з Γ істинна в даній інтерпретації.

Відмітимо деякі властивості формул при інтерпретаціях:

1. Формула A хибна в даній інтерпретації тоді і тільки тоді, коли $\sim A$ істинна в тій же інтерпретації, і A істинна тоді і тільки тоді, коли $\sim A$ хибна.

2. Жодна формула не може бути одночасно істинною і хибною в одній і тій же інтерпретації.

3. Якщо в даній інтерпретації істинні A і $A \longrightarrow B$, то істинне B.

4. $A \longrightarrow B$ хибне в даній інтерпретації тоді і тільки тоді, коли A в цій інтерпретації істинне, а B хибне.

5. (a) $A \wedge B$ виконується на послідовності s тоді і тільки тоді, коли A виконується на s і B виконується на s. $A \vee B$ виконується на s тоді і тільки тоді, коли A виконується на s або B виконується на s. $A \longleftrightarrow B$ виконується на послідовності s тоді і тільки тоді, коли або A виконується на s і B виконується на s, або коли A не виконується на s і B не виконується на s.

   Будемо казати, що інтепретація M даної теорії першого порядку *ізоморфна*

   (b) $(\exists x_i)A$ виконується на s тоді і тільки тоді, коли A виконується хоч би на одній послідовності s', яка відрізняється від s не більш ніж однією i-ю компонентою.

6. A істинне в даній інтерпретації тоді і тільки тоді, коли в цій інтерпретації істинне $(\forall x_i)A$. *Замиканням* даної формули A назвемо формулу, яка отримується приписуванням до A попереду знаків кванторів загальності, які містіть в порядку спадання індексів всі вільні предметні змінні, що входять в A.

7. Кожний частинний випадок всякої тавтології істинний у кожній інтерпретації.

8. Нехай вільні предметні змінні формули A містяться серед змінних $x_{i_1}, \ldots, x_{i_n}$. Тоді якщо у послідовностей s і s' компоненти з номерами $i_1, \ldots, i_n$ співпадають, то формула A виконується на s тоді і тільки тоді, коли вона виконується на s'.

9. Якщо формула A замкнена, то в довільній даній інтепретації або істинна A, або істинна $\sim A$ (тобто хибна A).

Формула A числення предикатів називається *загальнозначущою*, якщо вона істинна в кожній інтерпретації. Формула A числення предикатів називається *виконуваною*, якщо існує інтерпретація, в якій A виконується хоча б на одній послідовності із Σ.

Будемо казати, що інтепретація M даної теорії першого порядку $\mathbf{K}$ *ізоморфна* іншій інтерпретації M' теорії $\mathbf{K}$, якщо існує таке взаємно однозначне відображення g (називається *ізоморфізмом*) області D інтерпретації M на область D' інтерпретації M', що

(a) якщо $(A_j^n)^*$ і $(A_j^n)'$ — інтерпретації предикатної змінної A_j^n відповідно в M і M', то, які б не були $b_1, \ldots, b_n$ з D, умова $(A_j^n)^*(b_1, \ldots, b_n)$ виконується тоді і тільки тоді, коли виконується умова $(A_j^n)'(g(b_1), \ldots, g(b_n))$, тобто

$$(b_1, \ldots, b_n) \in (A_j^n)^* \longleftrightarrow (g(b_1), \ldots, g(b_n)) \in (A_j^n)';$$

(b) якщо $(f_j^n)^*$ і $(f_j^n)'$ — інтерпретації функціональної змінної f_j^n відповідно в M і M', то для довільних $b_1, \ldots, b_n$ з D справедлива рівність

$$g((f_j^n)^*(b_1, \ldots, b_n)) = (f_j^n)'(g(b_1), \ldots, g(b_n));$$

(c) якщо a_j^* і a_j' — інтерпретації предметної константи a_j відповідно в M і M', то $a_j' = g(a_j^*)$.

Відмітимо, що коли інтерпретації M і M' ізоморфні, то їх області мають однакову потужність, тобто однакове число елементів у випадку скінченних множин.

Теорема 30. *Якщо g — ізоморфізм інтерпретацій M і M', то (1) якими б не були формула A теорії $\mathbf{K}$ і послідовність $s = (b_1, b_2, \ldots)$ елементів області D, формула A виконується на s тоді і тільки тоді, коли вона виконується на відповідній послідовності $g(s) = (g(b_1), g(b_2), \ldots)$ і, отже, (2) формула A істинна в M тоді і тільки тоді, коли вона істинна в M'.*

Нехай $\mathbf{K}$ — теорія першого порядку, серед предикатних змінних є A_1^2. Будемо для скорочення писати $t = s$ замість $A_1^2(t, s)$ і $t \neq s$ замість $\sim A_1^2(t, s)$. Теорія $\mathbf{K}$ називається *теорією першого порядку з рівністю*, якщо наступні формули є теоремами $\mathbf{K}$:

(1) $(\forall x_1)(x_1 = x_1)$, (рефлексивність рівності);

(2) $(x = y) \longrightarrow (A(x, x) \longrightarrow A(x, y))$, (підстановочність рівності),

де x, y — предметні змінні, $A(x, x)$ — довільна формула, а $A(x, y)$ отримується з $A(x, x)$ заміною яких-небудь вільних входжень x входженнями y. Неважко тепер показати, що має місце таке твердження: *у всякій теорії першого порядку з рівністю*

(a) $\vdash t = t$ для довільного терма t;

(b) $\vdash x = y \longrightarrow y = x$;

(c) $\vdash x = y \longrightarrow (y = z \longrightarrow x = z)$.

Отже, для кожної моделі теорії першого порядку $\mathbf{K}$ з рівністю відношення ε, що відповідає в цій моделі предикатній змінні $=$, є відношенням еквівалентності. Якщо в області деякої моделі це відношення є тотожністю, то ця модель називається *нормальною*.

Нехай ω — кардинальне число. Теорія $\mathbf{K}$ першого порядку з рівністю називається *ω-категоричною* якщо 1) кожні дві нормальні моделі теорії $\mathbf{K}$, які мають потужність ω, ізоморфні, і 2) $\mathbf{K}$ має хоча б одну нормальну модель потужності ω.

3. Справедлива наступна теорема:

Теорема 31. *У кожному численні предикатів першого порядку всяка теорема є загальнозначуща формула.*

Справді, неважко бачити, що аксіоми, що задаються схемами (1) – (5) є загальноформули. Правила виведення MP і Gen зберігають властивість загальнозначущості, тому кожна теорема числення предикатів є загальнозначуща формула.

Теорема 32 (Гедель). *Кожна несуперечлива теорія першого порядку має зчисленну модель.*

Означення 7. *Теорія першого порядку* $\mathbf{K}$ *називається повною, якщо для довільної замкненої формули* A *теорії* $\mathbf{K}$ *або* $\vdash_{\mathbf{K}} A$*, або* $\vdash_{\mathbf{K}} \sim A$.

Теорема 33. *Кожна загальнозначуща формула несуперечливої теорії* $\mathbf{K}$ *першого порядку є теоремою теорії* $\mathbf{K}$.

Доведення. Достатньо розглянути лише замкнені формули A, оскільки кожна формула B загальнозначуща тоді і тільки тоді, коли загальнозначущим є її замикання, і виводиться в теорії $\mathbf{K}$ тоді і тільки тоді, коли в $\mathbf{K}$ виводиться її замикання. Отже, нехай A — загальнозначуща замкнена формула теорії $\mathbf{K}$. Припустимо, що A не є теоремою в $\mathbf{K}$. Тоді якщо ми додамо формулу $\sim A$ як нову аксіому до теорії $\mathbf{K}$, то отримаємо нову теорію $\mathbf{K}' = \mathbf{K} \cup \{\sim A\}$, яка буде несуперечливою.[14] Теорія $\mathbf{K}'$ має модель M (див. теорему 32). Оскільки $\sim A$ є аксіомою в $\mathbf{K}'$, то $\sim A$ істинна в M, а оскільки формула A загальнозначуща, то і вона істинна в M. Отже, ми прийшли до того, що формула A одночасно істинна і хибна в M, що неможливо. Таким чином, формула A повинна бути теоремою теорії $\mathbf{K}$. □

Наслідок 14 (теорема Геделя про повноту). *У кожному численні предикатів першого порядку теоремами є всі ті і тільки ті формули, які є загальнозначущими.*

[14] Припустимо, що $\mathbf{K}'$ суперечлива теорія, тоді знайдеться формула B така, що $\mathbf{K}' \vdash B \wedge \sim B$, тобто $\mathbf{K}, A \vdash B \wedge \sim B$, звідки $\mathbf{K} \vdash \sim A \longrightarrow B \wedge \sim B$. За законом контапозиції $\mathbf{K} \vdash \sim (B \wedge \sim B) \longrightarrow A$, тобто $\mathbf{K} \vdash \sim B \vee B \longrightarrow A$, але $\vdash \sim B \vee B$, тому $\mathbf{K} \vdash A$, що протирічить припущенню.

4.3 Формальна арифметика

Система аксіом формальної арифметики. Стандартна модель формальної арифметики, неповнота формальної арифметики.

1. Поряд з геометрією арифметика є найбільш безпосередньо інтуїтивною областю математики. Цілком природно тому саме з арифметики розпочати спробу формалізації й строгого обґрунтування математики. Перша напіваксіоматична побудова цієї дисципліни була запропонована Дедекіндом (1901 р.) і стала відомою під назвою "системи аксіом Пеано". Цю систему можна сформулювати таким чином:

(P_1) 0 є натуральне число;

(P_2) для довільного натурального числа x існує інше натуральне число, яке позначається x' і називається: (безпосередньо) слідуючим за x;

(P_3) $0 \neq x'$ для довільного натурального числа x;

(P_4) якщо $x' = y'$, то $x = y$;

(P_5) якщо Q є властивість, якою, можливо, володіють одні і не володіють інші натуральні числа, і якщо

1. натуральне число 0 володіє властивістю Q і
2. для кожного натурального числа x з того, що x володіє властивістю Q, випливає, що натуральне число x' володіє властивістю Q,

то властивістю Q володіють всі натуральні числа (*принцип індукції*).

Цих аксіом, разом з деяким фрагментом теорії множин, достатньо для побудови не тільки арифметики, але й теорії раціональних, дійсних та комплексних чисел. Однак в цих аксіомах містяться інтуїтивні поняття такі, як, наприклад, "властивість", що заважає всій системі бути строгою формалізацією. Тому ми зараз побудуємо деяку теорію першого порядку $\mathbf{S}$, засновану на системі Пеано, яка виявиться, при всій видимості, достатньою для виведення основних результатів елементарної арифметики.

Ця теорія першого порядку $\mathbf{S}$ буде мати тільки одну предикатну змінну A_1^2, одну предметну константу a_1 і три функціональних змінних f_1^1, f_1^2, f_2^2. Щоб не поривати із звичними нам в неформальній арифметиці позначеннями, в подальшому ми будемо писати $t = s$ замість $A_1^2(t, s)$, 0 замість a_1 і t', $t + s$, $t \cdot s$ відповідно замість $f_1^1(t)$, $f_1^2(t, s)$, $f_2^2(t, s)$, де t і s — терми. Наступні формули є *власними аксіомами теорії* $\mathbf{S}$:

(S_1) $x_1 = x_2 \longrightarrow (x_1 = x_3 \longrightarrow x_2 = x_3)$;

(S_2) $x_1 = x_2 \longrightarrow x_1' = x_2'$;

(S_3) $0 \neq (x_1)'$;

(S_4) $x_1' = x_2' \longrightarrow x_1 = x_2$;

(S_5) $x_1 + 0 = x_1$;

(S_6) $x_1 + x_2' = (x_1 + x_2)'$;

(S_7) $x_1 \cdot 0 = 0$;

(S_8) $x_1 \cdot (x_2') = (x_1 \cdot x_2) + x_1$;

(S_9) $A(0) \longrightarrow ((\forall x)(A(x) \longrightarrow A(x')) \longrightarrow (\forall x)A(x))$,

де $A(x)$ — довільна формула теорії $\mathbf{S}$.

Схема аксіом (S_9), яку ми будемо називати *принципом математичної індукції*, не відповідає повністю аксіомі (P_5) системи аксіом Пеано, оскільки в цій останній інтуїтивно припускаються $2^{\aleph_0}$ властивостей натуральних чисел, а схема аксіом (S_9) може мати справу лише із зчисленною множиною властивостей, які визначаються формулами теорії $\mathbf{S}$.

Аксіоми (S_3) і (S_4) відповідають аксіомам (P_3) і (P_4) системи аксіом Пеано. Аксіоми (P_1) і (P_2) пеанівської системи забезпечує існування нуля 0і операції "безпосередньо наступний", яким в теорії $\mathbf{S}$ відповідає предметна константа a_1 і функціональна змінна f_1^1. Аксіоми (S_1) і (S_2) забезпечують деякі необхідні властивості рівності, які Дедекіндом і Пеано передбачались як інтуїтивно очевидні. Аксіоми (S_5) – (S_8) являють собою рекурсивні рівності, які слугують означеннями операцій додавання й множення. Жодних постулатів, які відповідають цим аксіомам, Дедекінд і Пеано не сформулювали, тому що вони допускали використання інтуїтивної теорії множин, в рамках якої існування операцій $+$ і $\cdot$, які задовольняють аксіоми (S_5) – (S_8), вивідним.

2. Відмітимо, що інтерпретація теорії $\mathbf{S}$, в якій

(a) множина всіх цілих невід'ємних чисел слугує областю інтерпретації;

(b) ціле число 0 інтерпретує символ 0;

(c) операція взяття наступного (додавання одиниці) елемента інтерпретує функцію $'$ (тобто функціональну змінну f_1^1);

(d) звичайне додавання і множення інтерпретують як $+$ і $\cdot$;

(e) предикатна літера $=$ інтерпретується відношенням тотожності,

є нормальною моделлю теорії $\mathbf{S}$. Ця модель називається *стандартною моделлю* теорії $\mathbf{S}$. Кожна нормальна модель теорії $\mathbf{S}$, яка неізоморфна стандартній моделі, називається *нестандартною моделлю* теорії $\mathbf{S}$.

Якщо ми визнаємо цю стандартну інтерпретацію моделлю теорії $\mathbf{S}$, то тоді ми повинні будемо визнати і факт *несуперечності* цієї теорії. Однак семантичні методи, що включають в себе, як правило, відому долю теоретико-множинних міркувань, за думкою деяких математиків є дуже ненадійною основою для доведення несуперечності. Більш того, ми і не доводимо строго, що аксіоми теорії $\mathbf{S}$ істинні в стандартній інтерпретації, а приймаємо це твердження лише всього як інтуїтивно очевидне. Тому, а також з ряду інших причин прийнято кожний раз, коли на твердження про несуперечність теорії $\mathbf{S}$ спирається деяке доведення, явно посилатися на це твердження як на деяку недоведену гіпотезу.

3. В даній формальній теорії **S** немає жодних принципових перешкод для того, щоб кожну теорему, яка доводиться в курсах елементарної теорії чисел (наприклад, в книзі Виноградова И.М., Основы теории чисел), перевести на мову теорії **S** і побудувати виведення такого перекладу в цій теорії. Існують деякі теоретико-числові функції такі, як, наприклад, $x!$ і x^y, які можна визначити в **S**. З іншого боку, деякі класичні результати теорії чисел такі, як теорема Діріхле, доведені з допомогою теорії функцій комплексної змінної, причому часто невідомо навіть, чи можна отримати елементарні доведення (або виведення в **S**) для таких теорем. Деякі ж теореми теорії чисел (наприклад, теорема про прості числа) в самих формулюваннях містять неелементарні поняття, подібні до поняття логарифмічної функції, і такі теореми не можуть бути навіть сформульовані на мові теорії **S**, якщо тільки для них не існує еквівалентної елементарної форми.

Говорячи про силу і виражальні можливості теорії **S** розглянемо відому *теорему Геделя про неповноту формальної арифметики*. В цій теоремі доведено, що існують такі замкнені формули, які недовідні й неспростовні в теорії **S**, як тільки вона несуперечлива; отже, існує формула, істинна в стандартній інтерпретації, але невивідна в **S**. Виявляється, що така неповнота теорії **S** не може бути віднесена за рахунок нехватки яких-то істотних аксіом і що в основі цього явища криються більш глибокі причини, які діють також і у випадку інших теорій.

5 Елементи теорії алгоритмів

5.1 Поняття алгоритму та його характерні риси

Поняття про алгоритм, його характерні риси. Граф-схема алгоритму. Алфавіт. Кодування інформації. Алфавітні алгоритми.

1. Поняття *алгоритму* (або *алгорифму*) належить до основних понять сучасної математики. Воно поряд з іншими поняттями не визначається через більш прості поняття, а описується лише на прикладах. *Під алгоритмом в математиці прийнято розуміти скінченну систему точно визначених правил дії, виконання яких в певному порядку дає можливість розв'язувати всі задачі даного типу.* Сформульоване поняття алгоритму не є точним математичним означенням, оскільки у ньому не визначено, що треба розуміти під "правилами дії"і що таке "задачі даного типу". В сформульованому означенні тільки пояснюється зміст слова "алгоритм", в якому воно використовується в математиці. Це поняття зрозуміле кожному математику. Воно в загальній формі характеризує процеси, які в математиці задаються у вигляді словесних правил, формул, схем тощо. Правила дії, які дозволяють розв'язувати різні задачі, відомі з давніх часів і не тільки в математиці, але й в інших галузях людської діяльності.

Розглянемо два приклади відомих алгоритмів:

Приклад 1. *Алгоритм Евкліда.* Цей алгоритм використовується при знаходженні найбільшого спільного дільника для довільних двох даних натуральних чисел a і b. Нехай $a \geqslant b > 0$. Ділимо a на b, отримуємо остачу $r_1 < b$. Якщо $r_1 = 0$, то b є найбільший спільний дільник (НСД), якщо ж $r_1 > 0$, то шуканий дільник співпадає з НСД чисел b і r_1. Тому b ділимо на r_1, і якщо нова остача $r_2 = 0$, то r_1 — шуканий дільник, якщо $r_1 > r_2 > 0$, то ділимо r_1 на r_2. Продовжуючи цей процес, отримаємо:

$$a \geqslant b > r_1 > r_2 > \ldots > r_n > r_{n+1} = 0.$$

Останнє відмінне від нуля в цьому ланцюгу (r_n) і буде шуканим. При цьому весь процес закінчиться за скінченне число кроків, оскільки натуральних чисел, менших числа b, — скінченне число.

Приклад 2. *Додавання цілих чисел.* Вхідні дані — цифрові записи натуральних чисел у вигляді послідовності знаків: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, які називаються цифрами. Наприклад, 71, 935, 01026. Правила додавання можна сформулювати так:

1. Для фіксованих цифр доданків з врахуванням цифри перенесення записати відповідну їм цифру суми і відмітити нове перенесення (якщо воно є).

2. Фіксувати наступні вліво цифри доданків і місце нової цифри суми.

3. Повторити виконання правил 1 і 2, поки не будуть вичерпані всі значущі цифри доданків і перенесення.

Цей процес можна виконувати формально не задумуючись. *Кожний процес, який може бути автоматизований, є алгоритмічним процесом.*

З наведених прикладів можна зробити такі висновки:

1. Вхідні дані задач завжди задаються у вигляді деякої послідовності символів.

2. Процес розв'язування якої-небудь задачі являє собою послідовність перетворень записів вхідних даних в запис результату.

3. Послідовність перетворень складається з визначених елементарних актів (допустимих операцій), кожен з яких має формальний характер і може виконуватись автоматично не приймаючи до уваги змістовне значення записів вхідних даних.

4. Послідовність допустимих операцій, яка визначає деякий алгоритм, не залежить від конкретного виду вхідних даних.

5. Порядок виконання допустимих операцій визначається однозначно, тобто після виконання деякої допустимої операції точно відомим є наступний етап перетворень.

Аналіз змісту конкретних алгоритмів дозволяє сформулювати *основні загальні риси алгоритмів*:

1. *Визначеність алгоритму*, тобто однозначне визначення результату кожної допустимої операції і порядку виконання операцій. Виконання алгоритму являє собою строго детермінований процес, який не залежить від того, хто його виконує (обчислювач чи машина).

2. *Масовість алгоритму*, тобто можливість застосування алгоритму до різних вхідних даних. Алгоритм розв'язує не одну конкретну задачу, а визначений клас однотипних задач.

3. *Результативність алгоритму*, тобто скінченність процесу перетворення вхідних даних. Застосування алгоритму до вхідних даних завжди завершується утворенням певного результату — розв'язку задачі.

2. Інтуїтивне поняття алгоритму протягом довгого часу задовольняло математиків, доки термін алгоритм зустрічався в математиці лише в позитивних висловленнях типу "для розв'язування таких-то задач існує алгоритм і ось в чому він полягає". Теореми про неіснування алгоритмів не могли біти доведені за допомогою інтуїтивного поняття алгоритму в силу нечіткості цього поняття. Для доведення подібних теорем необхідно строге поняття алгоритму. В останній час рядом авторів розроблення теорії, які приводять до уточнення поняття алгоритму. Основою для одного з уточнень є теорія рекурсивних функцій, інші уточнення пов'язані з поняттям машини Тьюрінга і нормального алгоритму Маркова.

Числові функції, значення яких можна обчислити з допомогою застосування деякого алгоритму, називаються *обчислюваними функціями*. Оскільки поняття алгоритму в цьому означенні — інтуїтивне, то і поняття обчислювальної функції виявляється інтуїтивним. Дослідження показали, що сукупність частково обчислювальних функцій для самих різних розумінь алгоритму виявляється однією і тією ж. Всі часткові функції, алгоритми обчислення яких відомі, виявились *частково-рекурсивними*, тобто функціями, які визначаються певним чином з достатньою математичною строгістю. Кліні висунув гіпотезу про те, що клас алгоритмічно обчислювальних функцій співпадає з класом всіх частково-рекурсивних функцій. Раніше аналогічну гіпотезу відносно всюди визначених обчислювальних функцій висунув А. Черч. Гіпотези Черча і Кліні звичайно

об'єднують під назвою *тези Черча*. В силу тези Черча питання про обчислювальність функції рівносильний питанню про її рекурсивність. Поняття рекурсивності — строге, тому, у відомих випадках, можна довести, що розв'язуюча задачу функція не може бути рекурсивною, і, отже, алгоритм не може бути строго побудованим.

Постом і Тьюрінгом незалежно один від одного була висловлена одна думка, що процеси, які описуються алгоритмами, може здійснювати відповідна машина Тьюрінга. Тьюрінгом і Постом біли описані в точних математичних термінах класи машин, на яких можна здійснити або імітувати практично всі алгоритмічні процеси, які колинебудь описувались математиками. Такі машини називають в сучасний час *машинами Тьюрінга*. Дослідження показали, що клас функцій, в точності співпадає з класом всіх частково-рекурсивних функцій. Тим самим було отримане ще одне підтвердження тези Черча.

В алгоритмі Маркова (*нормальному алгоритмі*) вихідні дані для обчислювального процесу записуються у вигляді слова — послідовності літер. Обчислювальний процес зводиться до перетворення слів у відповідності з заданою програмою. Виявилось, що клас функцій, обчислювальних за допомогою нормальних алгоритмів, співпадає з класом частково-рекурсивних функцій. Таким чином, всі відомі уточнення поняття алгоритму призводять до одного і того ж класу функцій — частково-рекурсивним функціям. Це доводить еквівалентність перерахованих уточнень.

3. *Схема побудови алгоритмічної системи*:

1. Для описання задач задається система об'єктів (символів), між якими встановлюються деякі співвідношення. (*Алфавіт*)

2. Визначається сукупність *допустимих операцій.*

3. *Алгоритмічний процес* — процес застосування алгоритму до даного початкового стану.

4. *Алгоритм* визначається сукупністю допустимих операцій із вказівкою порядку їх виконання.

Граф-схемою алгоритму називається скінченна система точок (які називаються вузлами граф-схеми), які зв'язані між собою стрілками, що задовольняють такі властивості:

1. Існує точка, яка називається входом граф-схеми, з якої виходить лише одна стрілка і в яку стрілки не входять.

2. Існує точка, яка називається виходом граф-схеми, з якої не виходить жодна стрілка.

3. Інші вузли граф-схеми можуть бути або D-точками (дії), або P-точками (розпізнавання). З кожної D-точки виходить тільки одна стрілка. З кожної P-точки виходять дві стрілки: одна з міткою 1 ($1 \rightarrow$), інша з міткою 0 ($0 \rightarrow$). В кожну точку граф-схеми може входити довільне число стрілок.

4. *Абстрактним алфавітом* називається скінченна сукупність різних знаків (символів), які називаються *літерами* алфавіту. Позначаємо алфавіти $\mathfrak{A}$, $\mathfrak{B}$ тощо.

Слово — скінченна впорядкована сукупність літер даного алфавіту. Наприклад, якщо $\mathfrak{A} = \{a, b, c, d\}$, то a, aa, ab, abc, $abbadaadc$ — слова. Λ — *порожнє слово*. Два слова A і B називаються *рівними*, якщо вони складаються з однакових літер, які однаково розміщені, при цьому пишемо $A = B$. На множині слів вводиться операція *приписування слів*. Наприклад, якщо $A = abcd$, $B = bcda$, то $AB = abcdbcda$. Ця операція задовольняє такі властивості: $A(BC) = (AB)C$ і $\Lambda A = A\Lambda = A$ для довільних слів A, B, C. Таким чином, множина всіх слів $W_{\mathfrak{A}}$ в алфавіті $\mathfrak{A}$ утворює півгрупу з одиницею Λ відносно операції приписування слів.

Довжина слова — число літер в слові. Довжина слова A позначається через $l(A)$. Наприклад, якщо $A = abcda$, то $l(A) = 5$. $l(\Lambda) = 0$.

Слово A називається *початком слова* B, якщо існує слово C таке, що $B = AC$. Слово A називається *кінцем слова* B, якщо існує слово C таке, що $B = CA$. Слово A називається *підсловом слова* B, якщо існують такі слова C і D, що $B = CAD$.

Кодування інформації. Інформація про математичну задачу часто подається словами в алфавіті з 10-ти літер $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Однак натуральні числа можна подати в алфавіті з однієї літери $\{|\}$: 0 — Λ, 1 — $|$, 2 — $||$, 3 — $|||$, 4 — $||||$ тощо. Можна подати натуральні числа в алфавіті $\{0, 1\}$ у двійковій системі числення: 0 — 0, 1 — 1, 2 — 10, 3 — 11, 4 — 100, 5 — 101, 6 — 110, 7 — 111, 8 — 1000, 9 — 1001, 10 — 1010 тощо. В алфавіті $\{-, |, /\}$ можна зображати раціональні числа, наприклад, $-\frac{3}{5}$ зображується як $-|||/|||||$. В алфавіті $\{-, |, /, *\}$ можна зображати вектори: пара чисел $-\frac{3}{5}, \frac{1}{4}$ зображується словом $-|||/||||| * |/||||$. В алфавіті $\{-, |, /, *, \square\}$ можна зображати матриці. Наприклад, матриця $\begin{matrix} 2 & -3 \\ 4 & 2 \end{matrix}$ зображується словом $|| * -|||\square|||| * ||$. Отже, *всяка інформація про задачу може бути зображена словом в деякому абстрактному алфавіті.* Відмітимо, що можна будь-яку інформацію зображати лише дволітерному алфавіті $\{0, 1\}$, наприклад, 010, 0110, 01110, 011110 тощо.

Алфавітні алгоритми. Алгоритмічний процес розв'язування задачі можна розглядати як процес перетворення слів в абстрактному алфавіті. Вхідними даними і результатом алгоритму є слова.

Означення 8. *Алгоритмом в абстрактному алфавіті* $\mathfrak{A}$ *називається відповідність між словами в* $\mathfrak{A}$*, яка конструктивно задається скінченною системою допустимих операцій.*

Алгоритми будемо позначати через $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ тощо. Говорять, що *алгоритм* $\mathbf{A}$ *застосовний до слова* X, якщо процес перетворень вхідного слова X алгоритмом $\mathbf{A}$ закінчується деяким словом Y, при цьому записують $\mathbf{A}(X) = Y$. Сукупність слів даного алфавіту, до яких застосовний алгоритм $\mathbf{A}$, називається *областю застосовності алгоритму* $\mathbf{A}$. Алгоритм в розширені алфавіту $\mathfrak{A}$ називається *алгоритмом над алфавітом* $\mathfrak{A}$. Алгоритми $\mathbf{A}$ і $\mathbf{B}$ над алфавітом $\mathfrak{A}$ називаються *еквівалентними відносно* $\mathfrak{A}$, якщо для довільного слова X в алфавіті $\mathfrak{A}$, до якого застосовний алгоритм $\mathbf{A}$, застосовний також алгоритм $\mathbf{B}$, і результати їх дій співпадають, тобто $\mathbf{A}(X) = \mathbf{B}(X)$.

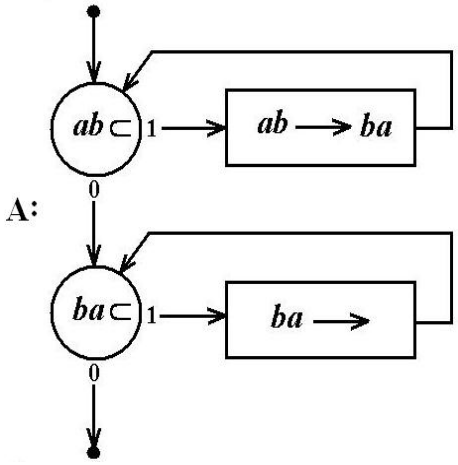
5.2 Нормальні алгоритми

Поняття про нормальний алгоритм. Узагальнений нормальний алгоритм. Нормальні алгоритми Маркова. Приклади нормальних алгоритмів. Принцип нормалізації. Операції над алгоритмами. Поняття про універсальний нормальний алгоритм.

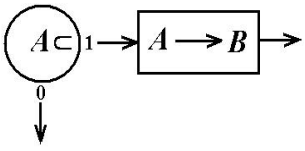
1. Нехай $\mathfrak{A}$ є деякий алфавіт. На множині слів над цим алфавітом $W_{\mathfrak{A}}$ розглянемо операцію підстановки слів, тобто операцію замінювання одного слова іншим, яка для довільних слів $A, B \in W_{\mathfrak{A}}$ позначається через $A \to B$ або $A \to \cdot B$, де стрілка $\to$ і крапка $\cdot$ не є літерами алфавіту $\mathfrak{A}$. Застосування операції підстановки до даного слова X зводиться до того, що перше (зліва) входження слова A в слово X замінюється словом B. Наприклад, якщо $X = YAZ$, то в результаті застосування підстановки $A \to B$ або $A \to \cdot B$ до X ми отримуємо слово $X_1 = YBZ$. Підстановка $A \to B$ називається *простою*, а $A \to \cdot B$ — *заключною*. Нехай $A \to (\cdot)\, B$ означає якусь одну з цих підстановок. Скінченний список підстановок в алфавіті $\mathfrak{A}$ називається *схемою алгоритму* і породжує, так званий, алгоритм $\mathbf{A}$ в алфавіті $\mathfrak{A}$, який позначається так:

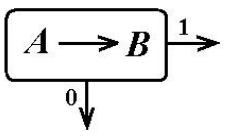
$$\mathbf{A}\colon \begin{cases} A_1 \to (\cdot)\, B_1, \\ A_2 \to (\cdot)\, B_2, \\ \cdots\cdots\cdots\cdots \\ A_r \to (\cdot)\, B_r. \end{cases}$$

Підстановка $A \to (\cdot)\, B$ не завжди застосовна до слова X, тому, щоб визначити, коли вона застосовна, а коли не застосовна, розглядають предикат $A \subset X$, значення якого дорівнює 1, якщо слово A входить в X, тобто є його підсловом, і дорівнює 0, якщо A не входить в X. Введемо тепер *операцію входження* даного слова A, яка позначається $A \subset$, результат застосування якої до слова X є значення предиката $A \subset X$. В *нормальних алгоритмах* єдиними допустимими операціями є операція підстановки та операція входження. *Нормальні алгоритми* визначаються послідовністю операцій входження та відповідних їм операцій підстановок із вказівкою порядку їх виконання. *Узагальнений нормальний алгоритм* в даному алфавіті $\mathfrak{A}$ задається граф-схемою, в якій D-точкам відповідають операції підстановки, а P-точкам — операції входження.

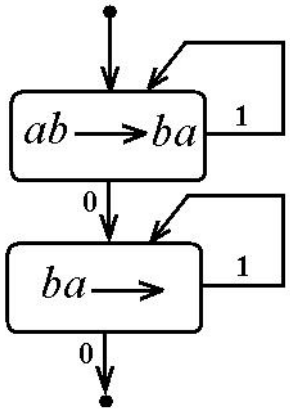


Приклад. На рисунку зображена граф-схема узагальненого нормального алгоритму в алфавіті $\{a, b\}$. Розглянемо слово $X = bbaba$ і застосуємо до нього алгоритм $\mathbf{A}$. Спочатку відмітимо входження слова ab у слові X. Маємо $X = bb\underline{ab}a$. Застосувавши першу підстановку до нього, отримуємо $X_1 = bb\underline{ba}a$. Застосуємо до X_1 другу підстановку, після чого матимемо $X_2 = b\underline{ba}$, звідки знову за другою підстановкою отримуємо слово $X_3 = b$, до якого жодна підстановка не застосовна.
Отже, $\mathbf{A}(bbaba) = b$.

Далі замість 

будемо малювати так: 

Тоді алгоритм **A** зобразиться так:



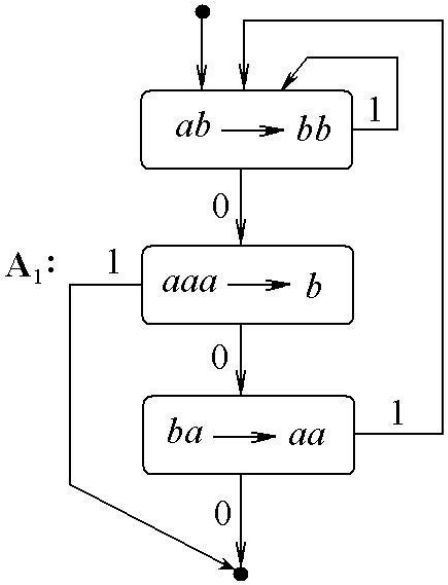
2. *Нормальні алгоритми А. А. Маркова* визначаються граф-схемами, які задовольняють умови:

а) Об'єднані вузли, упорядковані лінійно так, що стрілка з відміткою 0 одного об'єднаного вузла входить в наступний об'єднаний вузол. В перший об'єднаний вузол входить стрілка з початкової точки граф-схеми. З останнього вузла стрілка входить в заключну точку схеми.

б) Всі стрілки з відміткою 1 входять або в перший об'єднаний вузол, або в заключну точку граф-схеми.

Приклад 1. Розглянемо алгоритм, який заданий такою схемою підстановок:

$$\mathbf{A_1}: \begin{cases} ab \to bb, \\ aaa \to \cdot\, b, \\ ba \to aa. \end{cases}$$



Граф-схема цього алгоритму зображена на рисунку справа. Розглянемо слово $X = abba$ в алфавіті $\mathfrak{A} = \{a, b\}$ і застосуємо до нього алгоритм $\mathbf{A_1}$. Застосувавши до слова $X = \underline{ab}ba$ першу підстановку, ми отримаємо слово $X_1 = bb\underline{ba}$, звідки за третьою підстановкою матимемо $X_2 = bb\underline{aa}$. До слова X_2 застосовна лише третя підстановка, тому воно перетворюється у слово $X_3 = b\underline{aaa}$. До слова X_3 застосовні друга і третя підстановки, причому спочатку повинна виконуватись друга підстановка, але оскільки вона є заключною, то на цьому процес перетворення слів закінчується. Отже, слово X_3 переходить у слово $X_4 = bb$. Таким чином, $\mathbf{A_1}(abba) = bb$.

Приклад 2. Розглянемо тепер алгоритм $\mathbf{A_2}$, який задається такою схемою підстановок:

$$\mathbf{A_2}: \begin{cases} aba \to bb, \\ bb \to a, \\ aa \to b. \end{cases}$$

Застосувавши його до слова $X = ba\underline{bb}baa$, ми отримаємо наступну послідовність слів, кожне з яких отримується з попереднього слова за однією з підстановок алгоритму $\mathbf{A_2}$:

$X_1 = ba\underline{aba}a$, $X_2 = ba\underline{bb}a$, $X_3 = b\underline{aa}a$, $X_4 = \underline{bb}a$, $X_5 = \underline{aa}b$, $X_6 = b$. Отже, $\mathbf{A_2}(babbbaa) = b$.

Будемо казати, що *алгоритм* $\mathbf{A}$ *застосовний до слова* X, якщо про це перетворення слова X закінчується після *скінченного* числа кроків яким-небудь словом Y. В цьому випадку говорять, що алгоритм $\mathbf{A}$ *перетворює слово* X *в слово* Y і цей факт записують як $\mathbf{A}(X) = Y$. Якщо процес перетворення слова X алгоритмом $\mathbf{A}$ ніколи не закінчується то кажуть, що *алгоритм* $\mathbf{A}$ *не застосовний до слова* X.

Приклад 3. Алгоритм $\mathbf{A_3}$: $\{\rightarrow a$ не застосовний до жодного слова, оскільки процес приписування зліва літери a буде нескінченним. Зауважимо, що алгоритм $\mathbf{A_4}$: $\{\rightarrow \cdot a$ застосовний до будь-якого слова X. Він приписує зліва до X лише одну літеру a.

Приклад 4. В алфавіті $\{a, b\}$ нормальний алгоритм $\mathbf{A_5}$: $\{a \rightarrow aa$ не застосовний до слів, в які входить літера a, і застосовний до слів, які складаються лише з літер b.

Необхідною умовою можливості *побудови нормальних алгоритмів* , які реалізують який-небудь конструктивно заданий процес перетворення слів, є використання обох видів підстановок як звичайних, так і заключних.

Покажемо необхідність заключних підстановок. Нехай $\mathbf{A}$ є нормальний алгоритм, у якого немає заключних підстановок; X — слово, до якого застосовний алгоритм $\mathbf{A}$, тоді, очевидно, має місце рівність

$$\mathbf{A}(\mathbf{A}(X)) = \mathbf{A}(X), \tag{5.2.1}$$

тобто результат дії на слово $\mathbf{A}(X)$ є знову слово $\mathbf{A}(X)$, оскільки до нього вже не застосовні всі підстановки алгоритму $\mathbf{A}$. Умові (5.2.1) не задовольняє алгоритм $\mathbf{A_a}$, де $\mathbf{A_a}(X) = aX$, оскільки $\mathbf{A_a}(\mathbf{A_a}(X)) = \mathbf{A_a}(aX) = aaX$. Алгоритм $\mathbf{A_a}$, таким чином, не може бути реалізований нормальним алгоритмом зі схемами без заключних підстановок. В той же час ясно, що алгоритм $\mathbf{A_4}$: $\{\rightarrow \cdot a$ застосовний до кожного слова X.

По кажемо тепер, що неможливо обмежитись лише одними заключними підстановками. Дія нормального алгоритму з лише одними заключними підстановками, полягає з одноразового застосування однієї з підстановок. Тому довжина вихідного слова $\mathbf{A}(X)$ відрізняється від довжини вхідного слова X на скінченне число літер N, яка не залежить від довжини вхідного слова X. Число N визначається як максимум модулів різниці між довжинами слів в лівій і правій частинах підстановок алгоритму $\mathbf{A}$.[15] В той же час існують алгоритми, для яких різниця між довжинами вхідного і вихідного слів залежить від довжини вхідного слова і може бути як завгодно великою. Наприклад, для алгоритму подвоєння слів $\mathbf{A_{pod}}(X) = XX$. Цей алгоритм не може бути реалізований нормальним алгоритмом, що складається тільки із заключних підстановок.

3. Розглянемо деякі приклади нормальних алгоритмів.

Приклад 5. *Додавання натуральних чисел.* Алфавіт: $\{|, +\}$. Розглянемо нормальний алгоритм $\mathbf{A_1}$: $\{+ \rightarrow$ і продемонструємо як виконується дія $2 + 3 = 5$. За початкове слово візьмемо $X = || + |||$ і застосуємо до нього алгоритм $\mathbf{A_1}$, ми отримаємо слово $X_1 = |||||$. Отже, $\mathbf{A_1}(X) = X_1$.

---

[15] Тобто, якщо маємо алгоритм із заключними підстановками $\mathbf{A}$: $\begin{cases} X_1 \rightarrow \cdot Y_1, \\ \cdots\cdots\cdots \\ X_k \rightarrow \cdot Y_k, \end{cases}$ то $n_i = |l(X_i) - l(Y_i)|$, $i = 1, \ldots, k$, $N = \max(n_1, \ldots, n_k)$, а тому $|l(X) - l(\mathbf{A}(X))| \leqslant N$ для довільного слова X.

Приклад 6. *Віднімання натуральних чисел.* Алфавіт: $\{|, -\}$. Нормальним алгоритмом віднімання буде $\mathbf{A_2}$: $\begin{cases} |-| \to -, \\ - \to, \end{cases}$ і продемонструємо його роботу на прикладі $5-3=2$. За початкове слово візьмемо $X = |||||-|||$. Застосовуючи послідовно три рази першу підстановку, ми отримаємо такі слова: $X_1 = ||||-||$, $X_2 = |||-|$, $X_3 = ||-$. До слова X_3 застосовна лише друга підстановка, тому ми маємо $X_4 = ||$. Отже, $\mathbf{A_2}(X) = X_4$.

Приклад 7. *Множення двох натуральних чисел.* Алфавіт: $\{|, \times, a, b\}$. Нормальний алгоритм множення позначимо через $\mathbf{A_3}$, і він визначається такою системою підстановок:

$$\mathbf{A_3}: \begin{cases} b| \to |b, \\ a| \to |ba, \\ |\times \to \times a, \\ \times| \to \times, \\ \times \to, \\ a \to, \\ b \to |. \end{cases}$$

Продемонструємо дію алгоритму на такому прикладі: $2 \times 3 = 6$. За вихідне візьмемо слово $X = || \times |||$. До нього застосовна третя підстановка, тому слово X перетворюється в слово $X_1 = | \times a|||$. До слова X_1 в порядку розташування підстановок застосовна друга підстановка, тому X_1 перетворюється в слово $X_2 = | \times |ba||$, після чого за тією ж самою підстановкою X_2 перетворюється в слово $X_3 = | \times |b|ba|$. До X_3 вже застосовна перша підстановка, тому ми отримуємо слово $X_4 = | \times ||bba|$. До X_4 застосовна друга підстановка, тому ми далі матимемо $X_5 = | \times ||bb|ba$. Продовжуючи і далі подібні міркування, ми послідовно за тією чи іншою підстановкою отримуємо таку послідовність слів:

$$\begin{array}{lll} X_6 = | \times ||b|bba & X_{13} = \times||b|bbabbba & X_{20} = bbbbbb \\ X_7 = | \times |||bbba & X_{14} = \times|||bbbabbba & X_{21} = |bbbbb \\ X_8 = \times a|||bbba & X_{15} = \times||bbbabbba & X_{22} = ||bbbb \\ X_9 = \times|ba||bbba & X_{16} = \times|bbbabbba & X_{23} = |||bbb \\ X_{10} = \times|b|ba|bbba & X_{17} = \times bbbabbba & X_{24} = ||||bb \\ X_{11} = \times||bba|bbba & X_{18} = bbbabbba & X_{25} = |||||b \\ X_{12} = \times||bb|babbba & X_{19} = bbbbbba & X_{26} = |||||| \end{array}$$

Отже, ми показали, що $\mathbf{A_3}(X) = X_{26}$, тобто $\mathbf{A_3}(|| \times |||) = ||||||$, що означає $2 \times 3 = 6$.

Приклад 8. *Алгоритм скасування частини слова.* Такий алгоритм, який ми позначаємо через $\mathbf{A_4}$, визначається так: $\mathbf{A_4}(P\alpha Q) = P$ для довільних слів P, Q в алфавіті $\mathfrak{A}$, де $\alpha \notin \mathfrak{A}$. $\mathbf{A_4}$ задається такою системою підстановок: $\begin{cases} \alpha a \to \alpha, \\ \alpha \to, \end{cases}$ де $a \in \mathfrak{A}$.

Приклад 9. *Алгоритм подвоєння слів.* Розглянемо в алфавіті $\mathfrak{A} = \{a, b\}$, алгоритм подвоєння слів $\mathbf{A_5}(X) = XX$. Цей алгоритм реалізується в алфавіті $\{a, b, \alpha, \beta, \gamma\}$ схемою нормального алгоритму, що знаходиться праворуч. Нехай $X = bab$ є вихідне слово в даному алфавіті $\mathfrak{A}$, тоді $\mathbf{A_6}(bab) = babbab$. Процес перетворення вихідного слова алгоритмом буде такий:

$$\mathbf{A_5}: \begin{cases} ab\beta \to b\beta a, \\ aa\beta \to a\beta a, \\ ba\beta \to a\beta b, \\ bb\beta \to b\beta b, \\ \alpha a \to a\beta a\alpha, \\ \alpha b \to b\beta b\alpha, \\ \beta \to \gamma, \\ \gamma \to, \\ \alpha \to \cdot, \\ \to \alpha. \end{cases}$$

$X_1 = \alpha bab$, $X_2 = b\beta b\alpha ab$, $X_3 = b\beta ba\beta a\alpha b$, $X_4 = b\beta a\beta ba\alpha b$, $X_5 = b\beta a\beta bab\beta b\alpha$,

$X_6 = b\beta a\beta bb\beta ab\alpha$, $X_7 = b\beta a\beta b\beta bab\alpha$, $X_8 = b\gamma a\beta b\beta bab\alpha$, $X_9 = b\gamma a\gamma b\beta bab\alpha$,

$X_{10} = b\gamma a\gamma b\gamma bab\alpha$, $X_{11} = ba\gamma b\gamma bab\alpha$, $X_{12} = bab\gamma bab\alpha$, $X_{13} = babbab\alpha$, $X_{14} = babbab$.
Отже, $\mathbf{A_6}(X) = X_{14}$.

Приклад 9. *Алгоритм подвоєння слів в алфавіті двійників.* Алфавіт — $\mathfrak{A} = \{a, b\}$, алфавіт двійників — $\overline{\mathfrak{A}} = \{\overline{a}, \overline{b}\}$. Алгоритм подвоєнні слів в алфавіті двійників позначимо через $\mathbf{A_6}$. Якщо, наприклад, ми маємо слово $X = aba$, то то це слово в алфавіті двійників має вигляд $\overline{X} = \overline{a}\,\overline{b}\,\overline{a}$. Таким чином, $\mathbf{A_6}(X) = \overline{X}X$, тобто $\mathbf{A_6}(aba) = \overline{a}\,\overline{b}\,\overline{a}aba$. Розглянемо тепер процес перетворення слів цим алгоритмом слова ab. В результаті ми повинні отримати слово $\overline{a}\,\overline{b}\,ab$, тобто $\mathbf{A_6}(ab) = \overline{a}\,\overline{b}\,ab$. До слова $X = ab$ застосовна лише остання підстановка, тому ми отримуємо наступне слово $X_1 = \alpha ab$. На слово X_1 діє лише п'ята підстановка, за якою ми маємо $X_2 = \overline{a}\,a\alpha b$. За шостою підстановкою слово X_2 перетворюється в слово $X_3 = \overline{a}\,a\overline{b}\,b\alpha$. Далі за першою підстановкою отримуємо слово $X_4 = \overline{a}\,\overline{b}\,ab\alpha$, звідки за заключною сьомою підстановкою остаточно маємо слово $X_5 = \overline{a}\,\overline{b}\,ab$. Отже, $\mathbf{A_6}(X) = X_5$.

$$\mathbf{A_6}\colon \begin{cases} a\overline{b} \to \overline{b}a, \\ a\overline{a} \to \overline{a}a, \\ b\overline{a} \to \overline{a}b, \\ b\overline{b} \to \overline{b}b, \\ \alpha a \to \overline{a}a\alpha, \\ \alpha b \to \overline{b}b\alpha, \\ \alpha \to \cdot\,, \\ \to \alpha\,. \end{cases}$$

4. *Принцип нормалізації* полягає в тому, що для довільного конструктивно заданого алгоритму в деякому алфавіті $\mathfrak{A}$ можна побудувати еквівалентний йому нормальний алгоритм Маркова в деякому алфавіті, який є розширенням алфавіту $\mathfrak{A}$.

Цей принцип підтверджується експериментально. Впевненість в справедливості принципу нормалізації базується на всьому досвіді людства у створенні алгоритмів, оскільки серед різноманітних алгоритмів, які створені в наш час, немає таких, які не можуть бути нормалізовані. Принцип нормалізації можна сформулювати і так: *кожний алгоритм нормалізований.*

5. *Побудова нормальних алгоритмів.* Тепер ми розглянемо деякі операції, які можна виконувати над алгоритмами, що дають змогу за допомогою одних алгоритмів будувати інші.

5.1. *Композиція алгоритмів.* Послідовне застосування до даного слова X двох алгоритмів — спочатку $\mathbf{B}$, а потім $\mathbf{A}$ — називається *композицією* цих алгоритмів і позначається: $\mathbf{D}(X) = \mathbf{A}(\mathbf{B}(X))$, тобто $\mathbf{D} = \mathbf{AB}$. Побудова узагальненого алгоритму $\mathbf{D}$, який є композицією двох узагальнених нормальних алгоритмів $\mathbf{A}$ і $\mathbf{B}$, виконується просто. Необхідно вихідний вузол граф-схеми алгоритму $\mathbf{B}$ з'єднати з вхідним вузлом граф-схеми алгоритму $\mathbf{A}$.

5.2. *Об'єднання алгоритмів.* Алгоритм $\mathbf{D}$ називають *об'єднанням алгоритмів* $\boldsymbol{A}$ *і* $\boldsymbol{B}$, якщо він кожне слово X перетворює в об'єднання слів $\mathbf{A}(X)$ і $\mathbf{B}(X)$, тобто

$$\mathbf{D}(X) = \mathbf{A}(X)\mathbf{B}(X).$$

Для побудови граф-схеми об'єднання алгоритмів $\mathbf{A}$ і $\mathbf{B}$ треба переписати підстановки алгоритму $\mathbf{A}$ в алфавіті двійників і використати алгоритм $\mathbf{A_6}$ подвоєння слів. Заключною частиною алгоритму $\mathbf{D}$ є алгоритм $\mathbf{A_7}$ переходу від літер двійників до відповідних літер даного алфавіту (вважаємо, що $\mathfrak{A} = \{a, b\}$):

$$\mathbf{A_7}\colon \begin{cases} \overline{a} \to a\,, \\ \overline{b} \to b\,. \end{cases}$$

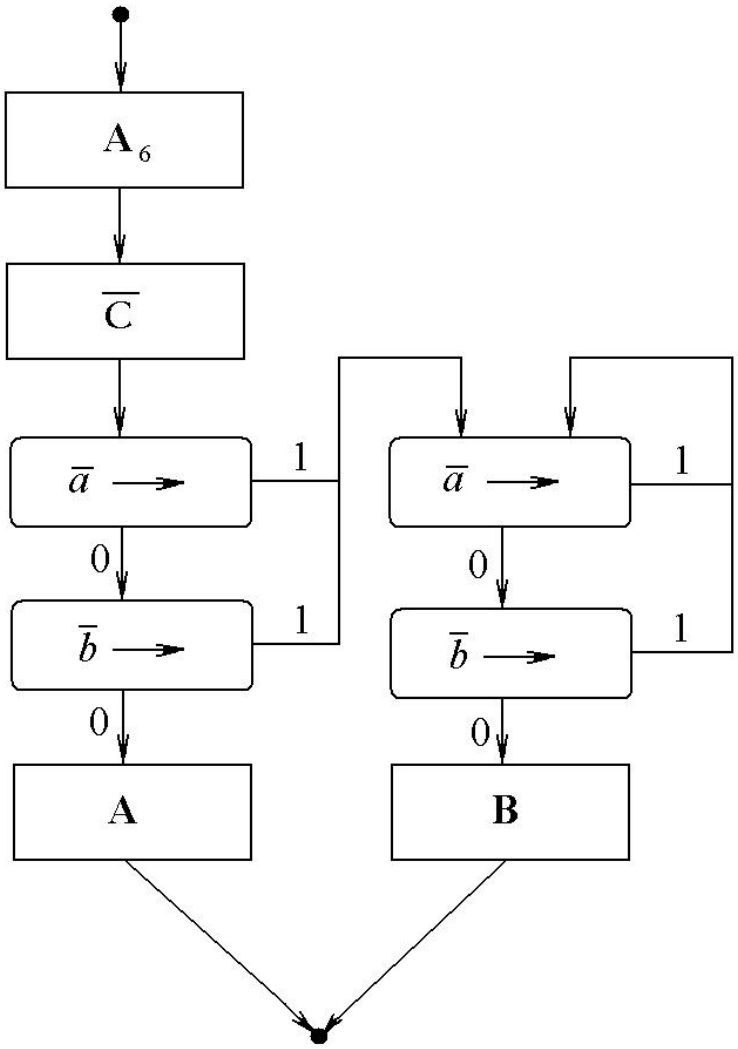
Отже, $\mathbf{D} = \mathbf{A_7}\overline{\mathbf{A}}\mathbf{B}\mathbf{A_6}$, де $\overline{\mathbf{A}}$ означає алгоритм $\mathbf{A}$ в алфавіті двійників.

5.3. *Розгалуження алгоритмів.* Алгоритм $\mathbf{D}$ називають *розгалуженням двох даних алгоритмів* $\boldsymbol{A}$ *і* $\boldsymbol{B}$*, який керується третім алгоритмом* $\mathbf{C}$, якщо для довільного слова

X в даному алфавіті $\mathfrak{A}$ маємо співвідношення:

$$\mathbf{D}(X) = \begin{cases} \mathbf{A}(X), & \text{якщо } \mathbf{C}(X) = \Lambda; \\ \mathbf{B}(X), & \text{якщо } \mathbf{C}(X) \neq \Lambda, \end{cases}$$

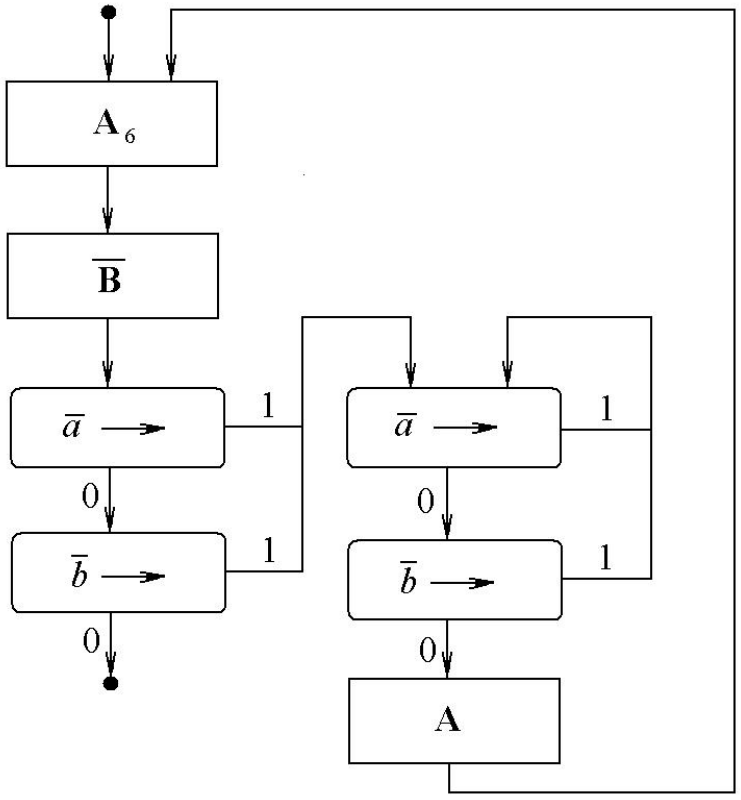
де Λ означає порожнє слово. Для побудови алгоритму $\mathbf{D}$ скористаємося алгоритмом $\mathbf{A_6}$ подвоєння слів і перепишемо підстановки алгоритму $\mathbf{C}$ в алфавіті двійників, тоді $\overline{\mathbf{C}}(\mathbf{A_6}(X)) = \overline{\mathbf{C}}(\overline{X})X$ для довільного слова X. Граф-схема розгалуження двох алгоритмів $\mathbf{A}$ і $\mathbf{B}$, який керується алгоритмом $\mathbf{C}$, має такий вид:



5.4. *Ітерація алгоритмів.* Багаторазове повторення даного алгоритму $\mathbf{A}$, яке керується алгоритмом $\mathbf{B}$, так що для довільного вхідного слова X існує послідовність слів $X = X_0, X_1, X_2, \ldots, X_n$ $(n \geqslant 0)$, для якої

$$\begin{aligned} &X_{k+1} = \mathbf{A}(X_k) \quad \text{при } 0 \leqslant k \leqslant n-1; \\ &\mathbf{B}(X_k) \neq \Lambda \qquad \text{при } 0 \leqslant k \leqslant n-1; \\ &\mathbf{B}(X_n) = \Lambda, \end{aligned}$$

називається *ітерацією алгоритму* $\mathbf{A}$*, яка керується алгоритмом* $\mathbf{B}$. Граф-схема ітерації має такий вигляд:



6. *Поняття про універсальний алгоритм.* Нехай $\mathbf{A}$ є деякий нормальний алгоритм в алфавіті $\mathfrak{A}$. Покажемо, що цей алгоритм зображується в алфавіті $\mathfrak{B} = \mathfrak{A} \cup \{\alpha, \beta, \gamma\}$, де $\{\alpha, \beta, \gamma\} \cap \mathfrak{A} = \varnothing$, одним словом. Для цього послідовно підстановки даного алгоритму виписуються так, що стрілки замінюються літерою α, крапка — літерою β, а слова підстановки розділяються за допомогою γ. Побудоване слово в алфавіті $\mathfrak{B}$ називається *зображенням* $\mathbf{A^u}$ *алгоритму* $\mathbf{A}$. Наприклад, розглянемо алгоритм

$$\mathbf{A}\colon \begin{cases} ab \to a, \\ aa \to \cdot\, b, \\ ba \to a \end{cases}$$

в алфавіті $\mathfrak{A} = \{a, b\}$. Тоді зображенням $\mathbf{A^u}$ цього алгоритму є слово:

$$ab\alpha a\gamma aa\alpha\beta b\gamma ba\alpha a.$$

Введемо нову літеру δ, тоді в алфавіті $\mathfrak{B}' = \mathfrak{B} \cup \{\delta\}$ будуємо зображення нормального алгоритму $\mathfrak{A}$ і вхідного слова X словом $\mathbf{A^u}\delta X$. Має місце наступна теорема, яку ми формулюємо без доведення:

Теорема. *Існує такий нормальний алгоритм* $\mathbf{N}$*, який називається універсальним нормальним алгоритмом, що для довільного алгоритму* $\mathbf{A}$ *і довільного вхідного слова* X *в алфавіті* $\mathfrak{A}$ *перетворює слово* $\mathbf{A^u}\delta X$ *в алфавіті* $\mathfrak{A} \cup \{\alpha, \beta, \gamma, \delta\}$ *в слово* $\mathbf{A}(X)$*, тобто* $\mathbf{N}(\mathbf{A^u}\delta X) = \mathbf{A}(X)$.

З цієї теореми випливає принципова можливість побудови такої машини, яка може виконувати роботу довільного нормального алгоритму.

5.3 Про алгоритмічно нерозв'язні проблеми

Асоціативне числення слів. Проблема еквівалентності слів. Приклади. Поняття про алгоритмічно нерозв'язні проблеми.

1. *Асоціативне числення слів.* В асоціативному численні слів вводяться допустимі операції підстановок без будь-яких обмежень на порядок їх застосування. Якщо $\mathfrak{A}$ — деякий алфавіт, а A і B — слова в ньому, то *неорієнтована* підстановка позначатиметься через $A \leftrightarrow B$. Наприклад, підстановка $ab \leftrightarrow bcb$ у двоелементному алфавіті $\mathfrak{A} = \{a, b\}$ може бути застосована до слова $abcbcbab$ в різному порядку, тобто можна в цьому слові виділяти (причому не обов'язково зліва) або слово ab, або слово bcb, тобто $\underline{ab}cbcbab$, або $a\underline{bcb}cbab$, або $abc\underline{bcb}ab$, або $abcbcb\underline{ab}$.

Означення 9. *Асоціативним численням називається сукупність всіх слів в даному абстрактному алфавіті разом з деякою скінченною системою допустимих підстановок. Асоціативне числення задається алфавітом і системою допустимих підстановок.*

Якщо слово A є результат одного застосування допустимої підстановки до слова B, то, очевидно, що слово B також є результатом застосування цієї ж підстановки до слова A; такі два слова називаються *суміжними словами*. Послідовність слів $A_1, A_2, \ldots, A_n$, в якій два сусідні слова є суміжними, називається *дедуктивним ланцюгом, який з'єднує слова A_1 і A_n*. Два слова A і B називаються *еквівалентними* (позначається через $A \simeq B$), якщо існує дедуктивний ланцюг, який з'єднує ці слова. Ясно, що $\simeq$ є відношення еквівалентності.

2. *Проблема еквівалентності слів* полягає в тому, що для довільних двох слів даного асоціативного числення необхідно визначити чи еквівалентні вони, чи ні. Ця проблема має важливе теоретичне і практичне значення в математиці.

Приклад 1. Асоціативне числення задається алфавітом $\{a, b, c\}$ і системою допустимих підстановок

$$ab \leftrightarrow ba,$$
$$ac \leftrightarrow ca,$$
$$bc \leftrightarrow cb.$$

Розглянемо нормальний алгоритм:

$$\mathbf{A_2}\colon \begin{cases} ba \to ab, \\ ca \to ac, \\ cb \to bc. \end{cases}$$

Результат застосування цього нормального алгоритму до довільного слова X в алфавіті $\{a, b, c\}$ є слово, у якого є всі літери слова X, але вони упорядковані так, що спочатку йдуть всі літери "a", за ними — всі літери "b", і потім "c". Наприклад,

$$\mathbf{A_2}(bacbaca) = aaabbcc.$$

Алгоритм $\mathbf{A_2}$ перетворює еквівалентні слова в даному асоціативному численні в рівні слова.

Приклад 2. Асоціативне числення задається алфавітом $\{a, b, c\}$ і системою допустимих підстановок

$$b \leftrightarrow acc,$$
$$ca \leftrightarrow accc,$$
$$aa \leftrightarrow \Lambda,$$
$$cccc \leftrightarrow \Lambda.$$

Розглянемо нормальний алгоритм:

$$\mathbf{A_3}: \begin{cases} b \to acc, \\ ca \to accc, \\ aa \to \Lambda, \\ cccc \to \Lambda. \end{cases}$$

Слова в даному алфавіті, які можна отримати застосуванням алгоритму $\mathbf{A_3}$ до слів в цьому ж алфавіті, назвемо *зведеними словами*. Покажемо, що зведеними словами в даному алфавіті можуть бути лише слова:

$$\Lambda,\ c,\ cc,\ ccc,\ a,\ ac,\ acc,\ accc.$$

Дійсно, алгоритм $\mathbf{A_3}$ знищує літери "b" в словах даного алфавіту; літера "a" завжди входить перед літерою "c" у зведених словах, причому не більше одного разу; літер "c", нарешті, не може бути більше трьох.

Покажемо далі, що довільні два зведених слова не еквівалентні між собою. Зразу ж відмітимо, що при побудові дедуктивного ланцюга можна не користуватись першою підстановкою. Справді, якщо в кожному слові дедуктивного ланцюга замінити літери "b" на слово acc, то отримаємо послідовність слів, у якій всі сусідні слова або суміжні, або просто рівні.

Оскільки друга, третя і четверта підстановки не змінюють парності літер "a" і "c", то жодне з перших чотирьох слів, в які літера "a" не входить, не еквівалентне жодному з чотирьох слів, що залишаються, в які входить літера "a" (непарне число разів).

Залишається тепер переконатися в нееквівалентності наступних пар слів:

$$\text{"}\Lambda\text{" і "}cc\text{"};\quad \text{"}c\text{" і "}ccc\text{"};\quad \text{"}a\text{" і "}acc\text{"};\quad \text{"}ac\text{" і "}accc\text{"}.$$

Неважко бачити, що з еквівалентності хоча б однієї пари випливає еквівалентність інших пар слів, в чому легко переконатися.

Індексом входження літери "a" в слово X називається число всіх входжень літери "c", які зустрічаються правіше літери "a". *Індексом слова X* називається сума індексів всіх входжень літери "a". Наприклад, в слові "$accac$" перша зліва літера "a" має індекс 3, друга — 1, індекс слова — 4

Покажемо, що слова "a" і "acc" не еквівалентні. Індекси цих слів однакової парності (0 і 2). Підстановки $aa \leftrightarrow \Lambda$ і $cccc \leftrightarrow \Lambda$ не змінюють парності індекса слова. Підстановка ж $ca \leftrightarrow accc$ змінює парність індекса слова. Справді, розглянемо два слова: $AcaB$ і $AacccB$. Індекси входжень "a" в слові A змінюється на 2; індекси входжень "a" в слові B не змінюється; індекс входження "a" між словами A і B змінюється на 3. Отже, в цілому індекс слова змінюється на *непарне число*.

Припустимо від супротивного, що слова "a" і "acc" еквівалентні, тобто існує дедуктивний ланцюг, який зв'язує ці слова; ланцюг побудований за допомогою підстановок:

$ca \leftrightarrow accc$; $aa \leftrightarrow \Lambda$; $cccc \leftrightarrow \Lambda$. Оскільки підстановка $cccc \leftrightarrow \Lambda$ змінює число входжень літери "c" на 4, а підстановка $aa \leftrightarrow \Lambda$ зовсім не змінює числа входжень "c", то для побудови дедуктивного ланцюга, яка зв'язує "a" і "acc", необхідно застосувати підстановку $ca \leftrightarrow accc$, причому непарне число разів. Але при цьому індекс слова змінюється на непарне число, що протиріччить однаковій парності індексів цих слів. Отже, слова "a" і "acc" не еквівалентні.

3. *Поняття про алгоритмічно нерозв'язні проблеми.* Існування алгоритмічно нерозв'язних проблем можна сформулювати так: *існують такі класи задач, для яких не існує єдиного нормального алгоритму їх розв'язання.* Наприклад, російський математик П. С. Новіков вперше встановив алгоритмічну *нерозв'язність проблеми тотожності в теорії груп*. Ще одним прикладом нерозв'язної проблеми в теорії алгоритмів є *проблема розпізнання самозастосовності алгоритмів.*

Пояснимо в чому полягає ця проблема. Розглядаємо нормальні алгоритми в алфавіті $\mathfrak{A}$, який складається с двох літер. Нехай $\mathbf{A^u}$ є слово в алфавіті $\mathfrak{A}$, яке зображує алгоритм $\mathbf{A}$.кщо алгоритм $\mathbf{A}$ застосовний до слова $\mathbf{A^u}$, то $\mathbf{A}$ називається *самозастосовним*, інакше він називається *несамозастосовним*. Існують алгоритми обох видів. Наприклад, тотожній алгоритм $\mathbf{A_1}(X) = X$ самозастосовний, а алгоритм $\mathbf{A}_a(X) = aX$ приписування літери — несамозастосовний.

Проблема розпізнавання самозастосовності алгоритмів полягає в тому, щоб знайти єдиний алгоритм, який би за схемою довільного нормального алгоритму $\mathbf{A}$ *встановив, чи самозастосовний алгоритм* $\mathbf{A}$*, чи ні.*

Згідно принципу нормалізації цю задачу достатньо розглядати лише для нормальних алгоритмів. Існування алгоритмічно нерозв'язних проблем означає, що при відшуканні алгоритму, який розв'язує ту чи іншу проблему, потрібно мати на увазі, що такий алгоритм може взагалі і не існувати. Тому разом зі спробами побудови алгоритму треба намагатися також довести його існування.

5.4 Обчислювальні функції

Числові функції, найпростіші числові функції. Перетворення функцій: підстановка, примітивна рекурсія, мінімізація. Примітивно-рекурсивні і частково-рекурсивні функції.

1. Функція, яка визначена і приймає значення на множині натуральних чисел, називається *числовою функцією*. Надалі ми будемо розглядати тільки такі функції.

Перша алгоритмічна система була побудована на основі зображення обчислювальних функцій. Відомо, що функція визначається відповідністю між елементами множини значень аргументу та елементами множини значень функції. При визначенні функції жодних обмежень на характер закону відповідності не накладається. Можливо, навіть, що за означенням аргументу неможливо знайти відповідне значення функції. Таке становище недопустиме для обчислювальних функцій. Отже, функція, для якої існує алгоритм обчислення її значень, називається *обчислювальною функцією*.

Серед обчислювальних функцій виділемо *найпростіші функції:*

(a) функція слідування $s(x) = x' = x + 1$;

(b) функція-константа $C_a^n(x_1, \ldots, x_n) = a$;

(c) функція тотожності $I_m^n(x_1, \ldots, x_n) = x_m$, де $1 \leqslant m \leqslant n$, $n = 1, 2, \ldots$

2. *Оператор підстановки* $\mathbf{S^{n+1}}$. Нехай f є n-місна числова функція, $f_1, \ldots, f_n$ — m-місні числові функції, тоді через $\mathbf{S^{n+1}}(f, f_1, \ldots, f_n)$ будемо позначати таку m-місну функцію $g(x_1, \ldots, x_m)$, яка визначається рівністю

$$g(x_1, \ldots, x_m) = f(f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m))$$

для довільних $x_1, \ldots, x_m \in \mathbb{N}$. Оператор підстановки визначений для функцій $f_1, \ldots, f_n$ з однаковими аргументами. При необхідності підстановки функцій з різним числом аргументів, ускладнення можна подолати шляхом введення фіктивних аргументів. Наприклад, функцію двох аргументів $\varphi(x_1, x_2)$ можна подати у вигляді:

$$\varphi(x_1, x_2) = \psi(x_1, x_2, x_3) = \varphi(I_1^3(x_1, x_2, x_3), I_2^3(x_1, x_2, x_3)).$$

Оператор примітивної рекурсії $\mathbf{R}$. Нехай g є n-місна, h — $(n+2)$-місна і f — $(n+1)$-місна числові функції. Будемо казати, що f *отримується з g і h за допомогою оператора примітивної рекурсії* (при цьому пишемо $f = \mathbf{R}(g, h)$), якщо для довільних $x_1, \ldots, x_n, y \in \mathbb{N}$ справедливі рівності:

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, y+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)).$$

Знайдемо послідовно значення числової функції f:

$$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n),$$
$$f(x_1, \ldots, x_n, 1) = h(x_1, \ldots, x_n, 0, g(x_1, \ldots, x_n)),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$f(x_1, \ldots, x_n, m+1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, m)).$$

Приклад. Нехай $g = 0$, $h = 2x+y$, тоді $f(0) = 0$, $f(1) = 0$, $f(2) = 2$, $f(3) = 2{\cdot}2{+}2 = 6$, $f(4) = 2 \cdot 3 + 2 \cdot 2 + 2 = 12$; $f(x+1) = 2x + 2(x-1) + \cdots + 2 \cdot 2 + 2 = 2\,\frac{x+1}{2}\,x = (x+1)x$.

Оператор мінімізації **M**. Цей оператор дає можливість визначати функцію від n аргументів за допомогою функції $(n{+}1)$-го аргументу. Нехай дана обчислювальна функція $g(x_1, \ldots, x_n, y)$; фіксуємо значення аргументів $x_1 = a_1, \ldots, x_n = a_n$. *Оператором мінімізації* визначається найменше натуральне число β, для якого $g(a_1, a_2, \ldots, a_n, \beta) = 0$; позначається дана операція так:

$$\beta = \mu_y[\, g(a_1, a_2, \ldots, a_n, y) = 0\,],$$

де μ_y є символ даної операції. Очевидно, що значення β є функцією від $a_1, a_2, \ldots, a_n$. Таким чином, ми маємо означення обчислювальної функції $f(x_1, x_2, \ldots, x_n)$ $(= (\mathbf{M}g)(x_1, x_2, \ldots, x_n))$ за допомогою операції μ_y (яка називається *операцією найменшого кореня*):

$$f(x_1, x_2, \ldots, x_n) = \mu_y[\, g(x_1, x_2, \ldots, x_n, y) = 0].$$

Якщо не існує таких значень y, при яких $g(x_1, \ldots, x_n, y) = 0$, то функція $f(x_1, \ldots, x_n)$ вважається невизначеною на відповідному наборі значень $x_1, x_2, \ldots, x_n$. Крім того, вважається, що функція $f(x_1, \ldots, x_n)$ невизначена на такому наборі значень $x_1 = a_1, \ldots, x_n = a_n$, для якого існує корінь рівняння $g(a_1, \ldots, a_n, y) = 0$, але хоча б для одного значення $(0 \leqslant \gamma < \beta)$ функція $g(a_1, \ldots, a_n, \gamma)$ невизначена.

3. Функція f називається *примітивно-рекурсивною*, якщо вона є однією з найпростіших функцій s, C_0^1, I_m^n або може бути отримана з найпростіших функцій за допомогою скінченного числа операторів підстановки і примітивної рекурсії. Відмітимо, що оператори підстановки та примітивної рекурсії, застосовані до всюди визначених функцій, дають також всюди визначені функції, тому всі примітивно рекурсивні функції всюди визначені.

Приклад. Покажемо, що n-місна функція константа є примітивно-рекурсивною функцією. Справді, це випливає з такої рівності:

$$C_a^n(x_1, \ldots, x_n) = \underbrace{s(s(\ldots s}_{a \text{ разів}}(C_0^1(I_1^n(x_1, \ldots, x_n)))\ldots)),$$

де $s(x) = x + 1$.

Функція f називається *частково-рекурсивною*, якщо вона може бути отримана з функцій s, C_0^1, I_m^n за допомогою скінченного числа операторів підстановки, примітивної рекурсії та мінімізації. З означення випливає, що кожна примітивно-рекурсивна функція є частково-рекурсивною. Тому клас частково-рекурсивних функцій включає в себе як підклас клас всіх примітивно-рекурсивних функцій.

Поняття частково- рекурсивної функції є одним з основних понять теорії алгоритмів. Які б класи алгоритмів до цього часу не будувались, у всіх випадках виявлялося, що числові функції, обчислювальні за допомогою цих алгоритмів, були частково рекурсивними. Тому загальноприйнятим є теза А. Черча відносно частково-рекурсивних функцій: *клас алгоритмічно обчислювальних числових функцій співпадає з класом всіх частково-рекурсивних функцій.*

На завершення відмітимо, що Гедель, використовуючи апарат частково-рекурсивних функцій, довів теорему про *неповноту формальної арифметики.*

5.5 Машина Тьюрінга

Означення машини Тьюрінга. Допустимі елементарні операції машини Тьюрінга. Програма роботи машини, конфігурація, функціональна схема машини Тьюрінга.

1. В зв'язку з розвитком сучасної обчислювальної математики особливий інтерес має така алгоритмічна система, в якій поняття алгоритму базується на командно-адресному принципі. Для наукового аналізу обчислювальних процесів, які можуть бути реалізовані машиною, бажано знайти просту за своєю логічною структурою схему алгоритмічної машини, яка може біти предметом точної математичної теорії. Вперше схему такої алгоритмічної машини побудував англійський математик А. Тьюрінг в 1937 році (до створення сучасних ЕОМ).

Ідея машини Тьюрінга базується на загальному аналізі процесів обчислення значень функцій обчислювачем. При цьому має місце *основна гіпотеза теорії алгоритмів* (теза Тьюрінга): *кожний алгоритм може бути реалізований в машині Тьюрінга.*

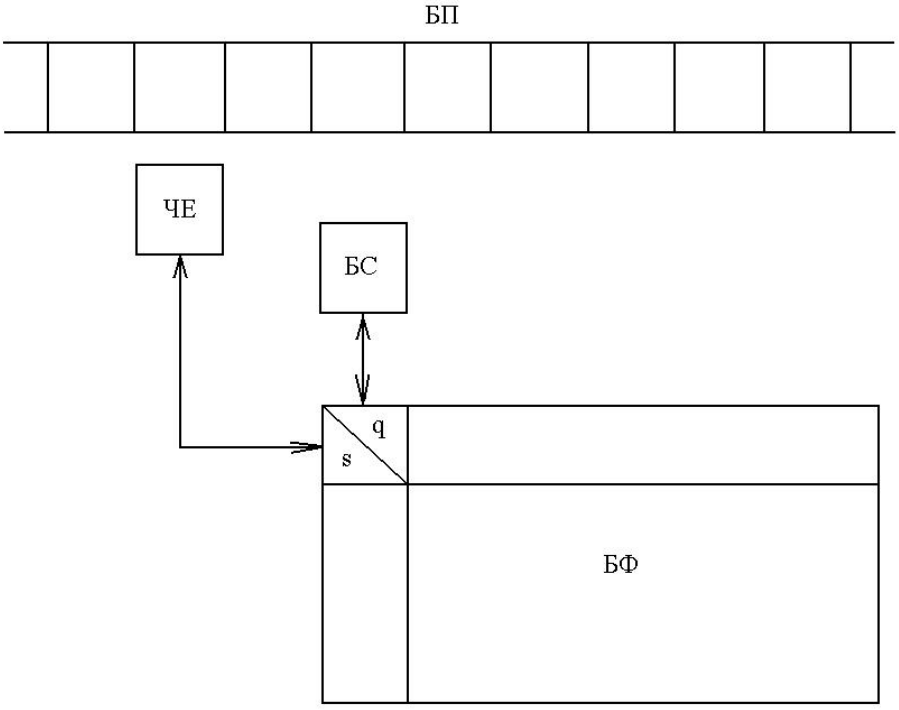
Означення машини Тьюрінга. Основними блоками машини Тьюрінга є:

1) *Блок зовнішньої пам'яті (БП).* Це нескінченна в обидва боки стрічка, яка розбита на комірки.

2) *Функціональний (програмний) блок (БФ).* Цей блок зображується скінченною таблицею з двома входами.

3) *Читаючий елемент (ЧЕ).* Він фіксує одну комірку стрічки пам'яті.

4) *Блок стану (БС) (блок внутрішньої пам'яті).* Він фіксує той чи інший стан машини.



Вхідна інформація подається (кодується) в комірках пам'яті літерами *зовнішнього алфавіту*: $\sigma = \{s_0, s_1, \ldots, s_k\}$. В кожній комірці може міститися лише одна літера. Серед літер алфавіту є літера, яка відповідає "порожньому символу", вважаємо її літерою s_0 або θ. "Порожній символ"знаходиться у всіх комірках пам'яті, які не зайняті іншими літерами зовнішнього алфавіту.

Стани машини Тьюрінга кодуються літерами *внутрішнього алфавіту*:

$$\mathcal{C} = \{q_0, q_1, \ldots, q_m\}$$

і визначаються змістом блоку стану. Серед станів машини Тьюрінга виділяється *заключний стан*, який означає завершення роботи машини (зупинка машини); позначають його символом "!".

Адреси комірок пам'яті машини Тьюрінга позначаються трьома літерами: Н, П, Л. Літерою Н позначається комірка, яку фіксує (оглядає) читаючий елемент. Літерою П позначається адреса комірки, яка знаходиться правіше від фіксованої комірки. Літерою Л позначається адреса комірки, сусідньої зліва від фіксованої комірки.

Допустимими елементарними операціями в машині Тьюрінга є такі три операції:

1) *Записування в фіксовану комірку пам'яті деякої літери зовнішнього алфавіту*. При цьому попередній вміст фіксованої комірки витирається.

2) *Перехід до нового стану*. При цьому в блок станів поміщається відповідна літера внутрішнього алфавіту.

3) *Фіксування комірки пам'яті за однією з адрес* Н, П, Л. При цьому здійснюється зміна розташування читаючого елемента.Точніше, за адресою Н розташування ЧЕ не змінюється, за адресою П читаючий елемент зсувається на одну комірку вправо, а за адресою Л — вліво.

Кожна *команда дії* машини Тьюрінга зображується трьома літерами:

$$sqC,$$

де s — літера зовнішнього алфавіту, яка записується у фіксовану комірку; q — літера внутрішнього алфавіту, яка розміщується в блок станів; C — одна з літер Н, П, Л. Команди розміщуються в комірках функціонального блоку, які за стовпцями помічені літерами станів, а за рядками — літерами зовнішнього алфавіту.

Програма роботи машини Тьюрінга задається командами дії, які розміщуються в комірках функціонального блоку. Окремий етап дії машини Тьюрінга складається з виконання деякої команди, яка міститься в комірці функціонального блоку, відміченого літерою станів, яка міститься у блоці станів, і літерою зовнішнього алфавіту, яка міститься у фіксованій комірці пам'яті.

Конфігурацією машини Тьюрінга називається зображення стрічки пам'яті з розміщеними на ній літерами зовнішнього алфавіту і відміченим станом машини фіксованої комірки пам'яті, яка фіксується читаючим елементом.

Функціональною схемою машини Тьюрінга називається функціональний блок з розміщеними в його комірках командами.

Починається робота машини Тьюрінга із задання початкової конфігурації і функціональної схеми машини.

Приклад. *Алгоритм додавання натуральних чисел* $3+2$.

Зовнішній алфавіт: $\sigma = \{\theta, 1, *\}$

Внутрішній алфавіт: $\mathcal{C} = \{q_0, q_1, q_2, !\}$

Початкова конфігурація машини Тьюрінга:

			1	**1**	**1**	*	**1**	**1**			

q_0

Функціональна схема машини Тьюрінга:

	$\mathbf{q}_0$	$\mathbf{q}_1$	$\mathbf{q}_2$
1	$\theta \mathbf{q}_2 \Pi$	$\mathbf{1q}_1 Л$	$\mathbf{1q}_2 \Pi$
θ	$\theta \mathbf{q}_0 \Pi$	$\theta \mathbf{q}_0 \Pi$	$\mathbf{1q}_1 Н$
*	$\theta !$	$*\mathbf{q}_1 Л$	$*\mathbf{q}_2 \Pi$

Література

[1] В. И. Игошин, Математическая логика и теория алгоритмов, Саратов: Изд. СГУ, 1991.

[2] Я. В. Хромой, Математична логіка, К.: Вища школа, 1983.

[3] Ф. М. Лиман, Математична логіка і теорія алгоритмів, Суми: Слобожанщина, 1998.

[4] Я. В. Хромой, Збірник вправ і задач з математичної логіки, К.: Вища школа, 1978.

[5] В. И. Игошин, Задачник-практикум по математической логике, М.: Просвещение, 1986.

[6] Р. Столл, Множества, Логика. Аксиоматические теории. М.: Просвещение, 1978.

[7] Э. Мендельсон, Введение в математическую логику, М.: Наука, 1971.

[8] Л. А. Калужнин, Что такое математическая логика, М.: Наука, 1964.

[9] П. С. Новиков, Элементы математической логики, М.: Физматгиз, 1959.

[10] С. К. Клини, Математическая логика, М.: Изд. "Мир", 1973.

[11] Л. А. Калужнін, В. С. Королюк, Алгоритми і математичні машини, К.: Радянська школа, 1964.

[12] Ю. Л. Ершов, Е. А. Палютин, Математическая логика, М.: Наука, 1979.

[13] А. Чёрч, Введение в математическую логику, т. 1, М.: Изд. иностр. лит., 1960.

[14] С. Г. Гиндикин, Алгебра логики в задачах, М.: Наука, 1972.

[15] А. И. Мальцев, Алгоритмы и рекусивные функции, М.: Наука, 1965.

[16] С. К. Клини, Введение в метаматематику, М.: Изд-во иностр. лит., 1957.

[17] А. А. Френкель, И. Бар-Хиллел, Основания теории множств, М.: Изд. "Мир", 1966.