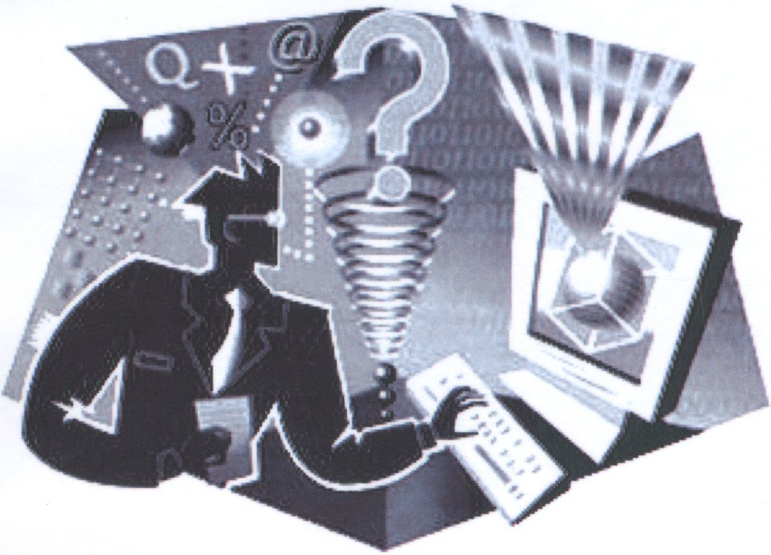
В. І. Месюра, Н. В. Лисак, О. І. Суприган



МАТЕМАТИЧНІ ОСНОВИ ЛОГІЧНОГО ПРОГРАМУВАННЯ

Міністерство освіти і науки України
Вінницький національний технічний університет

В. І. Месюра, Н. В. Лисак, О. І. Суприган

МАТЕМАТИЧНІ ОСНОВИ ЛОГІЧНОГО ПРОГРАМУВАННЯ

Вінниця
ВНТУ
2013

УДК 621.382
ББК 32.973–018:22.18
М53

Рекомендовано до друку Вченою радою Вінницького національного технічного університету Міністерства освіти і науки, молоді та спорту України (протокол № 9 від 21.04.2011 р.).

Рецензенти:
В. П. Кожем'яко, доктор технічних наук, професор, ВНТУ
А. М. Петух, доктор технічних наук, професор, ВНТУ
В. Т. Сусіденко, доктор економічних наук, професор, ВТЕІ КНТЕУ

Месюра, В. І.
М53 Математичні основи логічного програмування : навчальний посібник / В. І. Месюра, Н. В. Лисак, О. І. Суприган. – Вінниця : ВНТУ, 2013. – 94 с.

В посібнику розглянуті математичні основи логічного програмування: основні поняття логіки висловлювань та числення предикатів першого порядку, закони логіки, відношення логічного висновку, метод резолюцій. Посібник розроблений відповідно до плану кафедри та програми дисципліни "Функціональне та логічне програмування ".

УДК 621.382
ББК 32.973–018:22.18

ЗМІСТ

ПЕРЕДМОВА

Сучасний етап інформатизації суспільства характеризується широким впровадженням в усі сфери життєдіяльності людини інтелектуальних комп'ютерних технологій. Саме впровадження технологій штучного інтелекту дозволило перетворити комп'ютер зі звичайного потужного калькулятора, який просто швидко обробляв великі масиви числової інформації, на справжнього радника людини, здатного працювати із символьною інформацією і приймати обґрунтовані та ефективні рішення в областях, які вкрай важко піддаються формалізації.

Застосування комп'ютерів для розв'язання задач, які не мають алгоритмічних розв'язків, викликало необхідність розробки спеціальних інструментальних засобів, що повинні бути орієнтовані на знання і дозволяли б зручно моделювати процеси мислення людини.

Саме одним з таких засобів і є формальна логіка, яка забезпечує можливості формалізації процесу виведення на знаннях. Однак дійсно широке впровадження у техніці формальна логіка знайшла лише в комп'ютерних системах штучного інтелекту, завдяки зусиллям французьких вчених: після винаходу Робінзоном методу автоматичного доведення теорем, основаного на правилі резолюцій, та створення на його основі Кольмере і Руселем мови логічного програмування ПРОЛОГ (програмування у термінах логіки). Саме ПРОЛОГ був прийнятий за базову мову програмування в японській програмі створення комп'ютерів п'ятого покоління.

Однією з основних особливостей ПРОЛОГу є те, що він є декларативною мовою програмування. Тобто, мовою, яка, на відміну від традиційних, процедурних мов програмування (БЕЙСИК, ПАСКАЛЬ, СІ), описує не послідовність кроків (процедуру), які необхідно виконати для розв'язання задачі, а саме: подання задачі. На основі такого подання задача розв'язується автоматично, з використанням внутрішніх механізмів уніфікації та логічного виведення, реалізованих у середовищі ПРОЛОГу.

Метою даного навчального посібника є надання студентам систематизованої інформації з основ математичної логіки та логічного програмування, застосування логіки предикатів першого порядку для подання знань про предметне середовище, дії механізмів уніфікації термів та автоматичного доведення теорем на основі методу резолюцій, а також надання практичних навичок з формального описання задач у вигляді аксіоматичних систем, виявлення логічної структури задачі, кращого розуміння принципів програмування мовою логічного програмування PROLOG.

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

Даний навчальний посібник призначений для студентів напрямку "Комп'ютерні науки", які вивчають логічне програмування, а також може бути цікавим для широкого кола читачів, які відчули необхідність познайомитися з основами математичної логіки. Вивчення матеріалу не потребує будь-яких особливих початкових знань і допомагає оволодіти навичками аналізу причинно-наслідкових зв'язків, що існують у зовнішньому середовищі, які необхідні як при написанні програм, так і при виконанні будь-якої пізнавальної або наукової діяльності.

Структурно у посібнику можна виділити три основні частини, які присвячені питанням: логіки висловлень, числення предикатів першого порядку та автоматизації процесів логічного виведення.

У кожному розділі посібника містяться приклади, контрольні запитання та задачі, призначені для самостійної перевірки якості засвоєння матеріалу, а також посилання на літературні джерела, де можна отримати більш детальну інформацію з питань, що розглядаються.

У першому розділі пояснюється зв'язок, що існує між математичною логікою та логічним програмуванням, подано приклад написання програми узагальненою мовою логічного програмування.

Другий розділ присвячений розгляду основ логіки висловлень, знайомству з основними законами логіки та з'ясуванню такого фундаментального поняття логіки, як відношення логічного висновку.

У третьому та четвертому розділах показані недостатність логіки висловлень і здійснюється перехід до числення предикатів першого порядку, розглядаються правила побудови, інтерпретації та нормалізації правильно побудованих формул числення предикатів.

П'ятий розділ присвячений питанням автоматизації процедури логічного виведення за допомогою методу резолюцій, який є основою сучасних мов логічного програмування. Знайомство з принципом резолюції, правилами та загальним алгоритмом уніфікації, методом спростування на основі резолюцій, – дозволить майбутнім програмістам створювати більш досконалі програми мовами логічного програмування.

1 МАТЕМАТИЧНА ЛОГІКА І ЛОГІЧНЕ ПРОГРАМУВАННЯ

1.1 Основні етапи розвитку математичної логіки

Логічне програмування є порівняно новим напрямком в програмуванні та інформатиці, який оснований на ідеях та методах запозичених із математичної логіки. Саме слово логіка тлумачиться як "наука про закони мислення та його форми", або як "хід міркувань, умовиводів". Це слово походить від грецького слова "логос", що з одного боку означає "слово" або "мова", а з іншого – те, що виражає мова, тобто – мислення. Таким чином логіка має безпосереднє відношення до мови і стикається з граматикою та лінгвістикою.

Логіку створив видатний вчений стародавньої Греції – Арістотель в IV столітті до нашої ери. Він зробив це так вдало, що протягом багатьох століть вона залишалась незмінною. І в наш час розвиток логіки йде шляхом поширення та уточнення її основних положень, але не шляхом їх перегляду, як це сталося з більшістю інших наук (фізикою, геометрією). Але за часів Арістотеля логіка залишалась лише мистецтвом отримання логічних висновків з використанням природної мови і була занадто тісно пов'язана з самою природною мовою.

Лише в XVIII столітті німецький вчений Лейбніц вирішив створити нову логіку, яка стала би "мистецтвом числення". Він запропонував поставити у відповідність кожному поняттю деякий символ, і подати міркування у вигляді деяких обчислень. Однак його ідеї не були сприйняті вченими того часу і не отримали розвитку.

Ідею Лейбніца частково втілив у життя лише ірландський математик Дж. Буль, який в XIX столітті створив алгебру логіки, де діяли закони подібні до законів звичайної алгебри, але символами позначалися не числа, а речення. Мовою булевої алгебри можна було описувати міркування і "обчислювати" їх результати. Але вона охоплювала лише найпростіший тип міркувань. Алгебра логіки Буля на відміну від формальної логіки Арістотеля, отримала назву математичної логіки, оскільки вона використовувала мову та методи математики і була покликана до життя саме потребами математики.

На початку нашого сторіччя ряд вчених показали можливість застосування у техніці логіки висловлень, яка є одним з розділів математичної логіки. В середині сторіччя було виявлено тісний зв'язок математичної логіки з кібернетикою. З того часу розпочалось широке впровадження математичної логіки в медицині, біології, економіці, техніці та ін.

Наприкінці 50-х років розпочались роботи зі створення методу автоматизованого доведення теорем, в результаті яких Робінсону вдалося створити правило виведення, придатне для виконання машиною. Це правило Робінсон назвав резолюцією. Але пройшло ще більш ніж 10 років напруженої роботи багатьох вчених, доки французькі вчені Кольмороє і Русель у 1972 р. створили в Марсельскому університеті мову програмування, яка була основана на правилі резолюцій. Цю мову вони назвали ПРОЛОГ – мова ПРОграмування за допомогою ЛОГіки.

Перша версія ПРОЛОГу була створена на ФОРТРАНі, працювала дуже повільно і не знайшла широкого розповсюдження. Але ПРОЛОГ наочно продемонстрував великі потенційні можливості щодо реалізації ідей та методів штучного інтелекту, які були високо оцінені вченими різних країн. У 1981 році Міністерство зовнішньої торгівлі і розвитку Японії оголосило, що японський проект створення комп'ютерів п'ятого покоління буде основуватись саме на методі логічного програмування. Метою проекту було створення систем обробки інформації, що базуються на знаннях, мають можливість спілкування з користувачем за допомогою природної мові або графічних засобів, і можуть надавати експертні консультації під час розв'язання задач високого рівня. На відміну від традиційних, продуктивність Пролог-програм вимірюється за допомогою нового параметра – кількості логічних виведень за секунду. Сьогодні вже створені експериментальні зразки Пролог-машин, продуктивність яких складає більше за 1 Млвс/сек.

1.2 Поняття логічного програмування. Декларативні та процедурні мови програмування

У наш час ПРОЛОГ, разом з мовою функціонального програмування ЛІСП, відносять до основних мов штучного інтелекту, оскільки вони призначені перш за все для обробки символьної інформації. Але, як і ЛІСП, ПРОЛОГ має ряд переваг і при використанні в таких традиційних областях, як робота з текстами, базами даних, мережами та ін. Які ж переваги надає ПРОЛОГ порівняно з іншими мовами програмування?

Метою написання будь-якої програми є або розв'язання конкретної задачі, або подання системи, яка існує в реальному світі так, щоб програма могла моделювати її поведінку. Програмування за допомогою конструкцій деякої мови потребує мислення в термінах семантики цих конструкцій. Відповідно до природи семантики всі мови програмування розподіляють на три основні класи: процедурні, функціональні та реляційні або декларативні (рис.1).

Переважна більшість використовуваних у наш час мов програмування (СІ, Паскаль, Бейсик та ін.) відносяться до процедурних мов. Найбільш істотними класами декларативних мов є функціональні (ЛІСП, Лого, АПЛ і т.ін.) і логічні (Пролог, Пленер, Конівер тощо) мови.

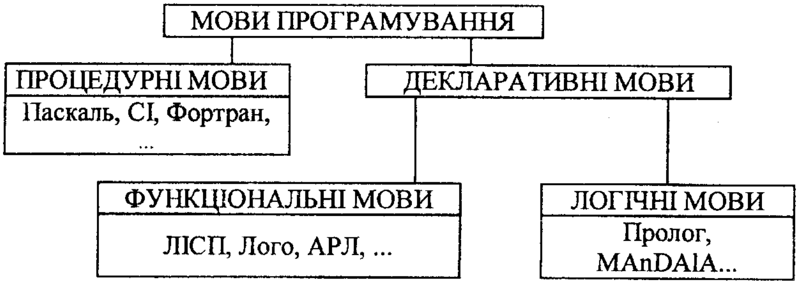


Рисунок 1 – Класифікація мов програмування

На практиці мови програмування не є чисто процедурними, функціональними або логічними, а містять у собі риси мов різних типів. Процедурною мовою часто можна написати функціональну програму або її частину і навпаки. Скоріше варто говорити не про тип мови, а про стиль або метод програмування. Природно, що різні мови підтримують різні стилі в різному ступені.

Переважна більшість сучасних комп'ютерів основана на моделі машини фон Неймана, яку було розроблено ще в 40-х роках. Машина фон Неймана складається з великої однорідної пам'яті, яка створюється з комірок і процесора, що має локальну пам'ять, комірки якої називаються регістрами. Процесор може виконувати операції завантаження даних з пам'яті в регістри, виконувати арифметичні і логічні операції над вмістом регістрів і відсилати дані з регістрів до пам'яті, а також виконувати додаткову множину команд керування, які впливають на вибір чергової команди в залежності від стану деякого регістра.

З вирішенням суто технічних проблем зі створення таких комп'ютерів на перший план стали виходити проблеми створення програмного забезпечення. Головним напрямком тут було створення мов програмування, які б спрощували роботу людини. І, хоча, ступінь абстракції мов від Асемблера до Паскаля і Ади постійно зростав, всі вони несуть на собі тягар архітектури фон Неймана, яка примушує людину міркувати термінами обмеженого набору операцій машини, що далеко не завжди є простою справою і часто потребує великих зусиль мислення. Однак формалізація задачі і методу розв'язання у вигляді набору інструкцій для машин фон Неймана дуже мало дає програмісту з точки зору поглиблення знань щодо розв'язуваної задачі. Висновком особливостей програмування на комп'ютерах з архітектурою фон Неймана є розподіл праці: є люди, які думають як розв'язати задачу і розробляють відповідні методи, а є люди – кодувальники, які пишуть тексти програм, тобто виконують прозаїчну та втомливу роботу перекладу інструкцій розробників в команди, які здатен сприймати комп'ютер. Оскільки за допомогою команд треба у явному вигляді задати послідовність дій для розв'язання задачі, тобто процедуру її розв'язання, мови програмування, які використовують для роботи на машинах фон Неймана, отримали назву процедурних мов програмування. Зміст конструкції процедурної мови визначається в термінах поведінки комп'ютера під час її виконання. Саме погляд на програмування як на процес "кодування" є глибинною причиною явища, яке називають "кризою програмування".

Однією з особливостей процедурних мов програмування високого рівня є також чітке розмежування даних та правил. Програма (процедура) – це свій замкнений світ, тобто жорстко зафіксований на етапі трансляції образ операцій з пасивними даними. Вплинути на процедуру можна тільки за допомогою параметрів.

Таким чином, процедурна програма складається з послідовності операторів і речень, що керують послідовністю їхнього виконання. Типовими операторами є оператори присвоювання і передачі управління, оператори введення-виведення і спеціальні речення для організації циклів. З них можна складати фрагменти програм і підпрограми. В основі процедурного програмування лежать узяття значення змінної, здійснення над ним дії і збереження нового значення за допомогою оператора присвоювання, і так доти поки не буде отримане (і, можливо, надруковане) бажане остаточне значення.

В логічному програмуванні необхідність відмежування даних від програми відсутня, оскільки сама програма може трактуватись як сукупність логічних правил. При цьому формалізація задачі полягає в описанні її мовою логіки, тобто у поданні в явному вигляді знань про предметну область та цілі розв'язання задачі, а не у поданні набору операцій деякого комп'ютера. Логіка дає точну мову для виведення висновків з вихідних положень, дає можливість, базуючись на істинності або хибності одних тверджень, робити висновки про істинність або хибність інших тверджень. Вона дозволяє обґрунтовувати несуперечність тверджень і перевіряти істинність доводів. Все це збагачує людину в процесі програмування знаннями про задачу, яку треба розв'язати.

Таким чином, логічне програмування основується на логічній моделі, що не пов'язана з деяким конкретним типом машинної моделі. Головна ідея логічного програмування полягає в тому, що не людину необхідно навчати мисленню в термінах операцій комп'ютера, а комп'ютер повинен виконувати інструкції, які відповідають образу мислення людини. У найбільш чистому вигляді логічне програмування передбачає, що самі інструкції взагалі не задаються, а замість цього явно, у вигляді логічних аксіом, формулюються відомості про задачу і припущення, яких досить для її розв'язання. Така програма ініціалізується на виконання при постановці задачі, яка формулюється у вигляді логічного твердження, яке потрібно довести. Таке твердження називається цільовим твердженням.

Виконання програми полягає у спробі розв'язати задачу, тобто довести цільове твердження з використанням припущень, які сформульовані у вигляді логічної програми.

Самі логічні мови програмування відносяться, таким чином, до категорії реляційних мов, оскільки зміст їх конструкцій визначається як відношення між окремими сутностями або класами сутностей. Тобто, реляційну конструкцію можна інтерпретувати як формулювання або декларацію того, що існує відношення між аргументами, які подані ім'ям цього відношення. Таким чином, під час використання логічних мов програмування не створюється описання процедури або послідовності кроків розв'язання задачі, а декларується система відношень між об'єктами задачі. Тому такі мови називають декларативними мовами програмування.

Функціональні мови програмування, найбільш відомим представником яких є ЛІСП, з точки зору розмежування даних та програми, знаходяться між процедурними та логічними мовами. Їх особливість полягає у використанні єдиної форми подання програми і даних. Конструкції мови мають дві інтерпретації, які залежать від способу їх використання. Тобто, є можливість трактування програми як даних або інтерпретації і застосування даних як програми. Використання і інтерпретацію виразів у функціональному програмуванні, тобто перетворення програми в дані і даних в програму, виконує програміст.

1.3 Основні конструкції логічного програмування

Логічна програма – це множина аксіом і правил, які виражають інформацію про задачу шляхом задання відношень між об'єктами. Обчисленням логічної програми є виведення з неї висновків. Тобто, програма задає множину висновків, які і є її значеннями.

Основне призначення ПРОЛОГу полягає в обробці символьної нечислової інформації. Особливо зручно використовувати його для розв'язання задач, в яких можна виділити об'єкти і відношення, які існують між ними. Особливістю ПРОЛОГу є однаковість програм і даних, які є лише двома різними точками зору на його об'єкти. В єдиній базі даних можна вільно створювати і знищувати окремі елементи. Оскільки різниця між програмами і даними відсутня, з'являється можливість модифікації програми безпосередньо в процесі її роботи.

Основними конструкціями логічного програмування є терми і твердження, які запозичені з логіки. Є три основних види тверджень: факти, правила і питання. Є одна єдина структура даних – логічний терм.

Терм – це або константа, або змінна, або функція. Під час використання функцію записують у вигляді символьного позначення функції, за яким у дужках розміщується список аргументів. Кожен аргумент в свою чергу сам є термом. Загальна форма застосування функції має такий вигляд:

$$f(t_1,t_2,...,t_n),$$

де $t_1,t_2,...,t_n$ – терми.

Функція подає відображення n елементів з області інтерпретації на один елемент цієї області. Наприклад, для області знань "арифметика" і області інтерпретації "множина натуральних чисел" важливою функцією є "множення", що відображує, наприклад, числа 2 та 3 на одне число 6, яке має назву "значення" даного використання функції множення.

Відношенням називають відображення n елементів з області інтерпретації на один з двох елементів множини {істина, хибність}. Для області знань "арифметика" і області інтерпретації "множина натуральних чисел" важливим відношенням є, наприклад, відношення "більш", яке має два аргументи. Так: пара чисел 2 і 3 відображується цим відношенням на значення "хибність", а пара чисел 3 і 2 – на значення "істина".

Розглянемо приклад використання логіки для розв'язання проблем. Дівчина Марія пішла до жіночого монастиря. Абатиса повідомила її про правила перебування в монастирі, зокрема про заборону пускати до монастиря чоловіків. Але одного разу абатиса помітила, що Марія пропустила до монастиря одного чоловіка. У своє виправдання Марія сказала, що не могла заборонити чоловікові вхід, оскільки його мати є свекрухою її матері. Абатиса погодилась, що даний випадок є винятком з правил, вирішивши, що Марія пустила до монастиря батька. Але існує і інше розв'язання.

Для подання проблеми побудуємо семантичну мережу родинних відносин (рисунок 2), вузли якої відображують об'єкти, а гілки – відношення між ними.

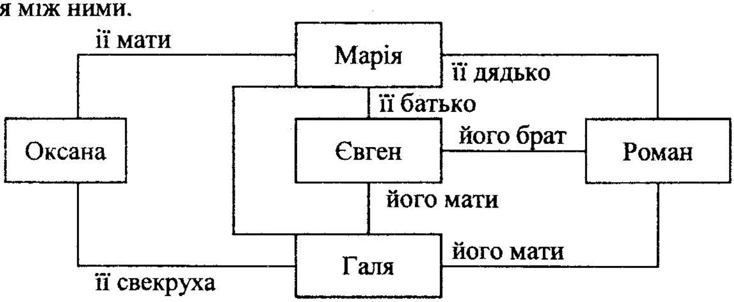


Рисунок 2 – Семантична мережа родинних відносин

Звідси вже можна побачити, що гостем може бути не тільки батько але й його брат, дядько Марії. Дана мережа відображає лише те вузьке навколишнє середовище, яке необхідно враховувати для розв'язання даної конкретної проблеми. Для того ж, щоб працювати з будь-якими родинними відносинами її треба суттєво доповнити.

Мовою логіки предикатів опис відповідного навколишнього середовища буде мати такий вигляд.

Перша група предикатів буде описувати факти, зображені семантичною мережею:

(1) мати (Оксана, Марія)	(передумова)
(2) свекруха (Галя, Оксана)	(передумова)
(3) мати (Галя, Євген)	(передумова)
(4) батько (Євген, Марія)	(передумова)
(5) мати (Галя, Роман)	(передумова)
(6) дядько (Роман, Марія)	(передумова)

Власно це і буде логічною програмою для прикладу, що розглядається.

Для ініціалізації програми на виконання введемо до системи цільове твердження – запит про те, що існує деякий чоловік (позначимо його через X), мати якого (позначимо її через Z) є свекрухою матері Марії (мати Марії позначимо через Y). Таке складене твердження можна подати трьома простими реченнями, які треба об'єднати між собою логічними зв'язками "і":

"Існує мати Марії на ім'я Y" і
"Існує жінка на ім'я Z, яка є свекрухою матері Марії Y" і
"Існує чоловік X, мамою якого є Z".

Мовою логіки предикатів таке цільове твердження можна записати у вигляді трьох простих фактів (трьох підцілей загальної цілі):

(7) мати (Y, Марія) і
свекруха (Z, Y) і (передумова)
мати (Z,X)

Кожний з цих фактів ПРОЛОГ має довести (підтвердити наявність даного факту в програмі) або спростувати в процесі виконання програми

Введення цільового запиту ініціалізує програму на виконання. При цьому Пролог починає послідовно переглядати всі факти в програмі (зверху – вниз), шукаючи в ній перший факт – мати (Y, Марія).

Під час перегляду вже першого рядка програми (1) ПРОЛОГ виявить факт – мати (Оксана, Марія), який повністю збігається з потрібним фактом мати (Y, Марія) в разі підстановки до змінної Y імені (значення константи) Оксана.

Отже, першу підціль буде задоволено підстановкою на місце змінної Y імені Оксана з першого (1) факту програми. При цьому три наші підцілі приймуть такий вигляд:

мати (Оксана, Марія) і
свекруха (Z, Оксана) і
мати (Z, X)

Тепер Пролог знов починає переглядати всі рядки програми з першого до останнього в пошуках факту, який би задовольняв другу підціль. Такий факт буде знайдено у другому рядку програми (2):

свекруха (Галя, Оксана).

Отже, ПРОЛОГ виконає заміну змінної Z на константу Галя, внаслідок чого другу підціль буде задоволено, а цільове твердження прийме такий вигляд:

мати (Оксана, Марія) і
свекруха (Галя, Оксана) і
мати (Галя, X)

При перегляді фактів програми ПРОЛОГ знайде факт

мати (Галя, Євген)

внаслідок чого змінній X буде дано значення Євген.

Оскільки в програмі було знайдено факти, що задовольняють всі три підцілі, ПРОЛОГ видасть повідомлення YES, яке свідчить про правильність зробленого у цільовому твердженні припущення. Крім того, він виведе значення змінних, при яких задовольняється цільове твердження:

X = Євген; Y = Оксана; Z = Галя.

Отже, внаслідок виконання програми було з'ясовано, що гостем Марії може бути Євген, тобто її батько. Але на етапі доведення останньої підцілі існує можливість і іншого розв'язку, оскільки для предиката мати (Галя, X) існує ще одна підстановка:

мати (Галя, Роман)

Тобто, цілком можливо, що в гості до Марії приходив її дядько Роман.

Формально-логічні передумови (1)...(6) називають фактами, які описують певну область навколишнього середовища. Передумова (7) – це питання або ціль, істинність якого підтверджується або спростовується за допомогою аналізу описання навколишнього середовища.

1.4 Контрольні запитання

1. Розшифруйте поняття, яке визначає термін “логіка”.
2. Чим математична логіка відрізняється від формальної логіки Арістотеля?
3. Назвіть основні, на Вашу думку, області застосування математичної логіки.
4. Які основні риси повинні були мати комп’ютери п’ятого покоління, що розроблялися в межах проекту Міністерства торгівлі і розвитку Японії?
5. Назвіть основні риси, які характеризують архітектуру комп’ютера фон Неймана.
6. Що таке процедурна мова програмування?
7. Наведіть основні відмінності логічних і процедурних мов програмування.
8. Які основні конструкції логічного програмування Вам відомі?
9. Які види тверджень використовуються у логічному програмуванні?
10. Що таке терм?
11. Наведіть означення функції і предиката.
12. Що таке функціональне програмування?

2 ФОРМАЛЬНІ СИСТЕМИ

2.1 Поняття формальної системи

У формальній логіці розробляються методи правильних міркувань, які являють собою ланцюжки умовиводів в логічно послідовній формі. Міркування про неї вивчаються з точки зору форми, а не змісту. З цією метою в звичайних міркуваннях виділяють певні елементи, які можна довільно замінювати будь-якими іншими елементами.

Розглянемо, наприклад, відому фразу (силогізм):

«Людина смертна, Сократ – людина, отже, Сократ смертний».

Не важко побачити, що тут імена класів «людина», «смертний» і «Сократ» можна замінити будь-якими іншими іменами і при цьому саме міркування залишиться формально правильним. Дійсно, у наведеному прикладі абстрактна модель розглянутого судження набуде такого вигляду:

«Якщо всі X суть Y і якщо Z є X, то Z є Y».

Таким чином вже просте заміщення імен класів в міркуваннях на символьні позначення дозволяє будувати узагальнені судження на основі подібних міркувань. Символи, які можна замінити називають змінними, а символи, які повинні бути фіксованими (наприклад, якщо, тоді, або, і, оскільки, Отже, є, належить, випливає та ін.), тобто які не можна замінити, називають операторами.

Формалізація процесів логічного виведення традиційно здійснюється за допомогою так званих формальних систем. Формальна система, це сукупність чисто абстрактних об'єктів (не пов'язаних із зовнішнім світом), в якій подано правила оперування множиною символів у чисто синтаксичному розумінні без будь-якого урахування змісту (семантики).

Формальну систему (ФС) можна визначити як четвірку:

$$Ф = \langle T, L, Q, R \rangle,$$

де T – кінцевий алфавіт (кінцева множина символів).

Тобто, T є множиною базових елементів, цеглин, які не можна роздробити на більш дрібні. Прикладами таких елементів можуть бути букви (графеми) або деталі дитячого конструктора. Єдина вимога до елементів множини T полягає в тому, що для будь-якого елемента за кінцеву послідовність кроків необхідно уміти визначити, чи належить він до T чи ні, а також уміти відрізняти кожен елемент від інших і ототожнювати однакові елементи;

L – певна процедура побудови формул (або слів) ФС. Тобто, L – це множина синтаксичних правил, за допомогою яких з елементів множини T будуються більш складні синтаксично правильні новоутворення. Наприклад, з букв утворюють лінійно впорядковані сполучення, які називають словами, реченнями, текстами; з деталей дитячого конструктора будують певні моделі;

Q – деяка підмножина формул, які називають аксіомами. Q містить виділені з якихось міркувань синтаксично правильні утворення.

R – скінченна множина правил виведення, які дозволяють отримати з деякої скінченної множини формул, деяку іншу множину формул. Такі правила в загальному вигляді можна записати так:

$$U_1 \text{ і } U_2 \text{ і } \ldots \text{ і } U_p \rightarrow W_1 \text{ і } W_2 \text{ і } \ldots \text{ і } W_n,$$

де U_i та W_i – формули ФС, а стрілка $\rightarrow$ позначає «спричиняє» або «випливає».

Алфавіт, який заздалегідь вважається скінченним, іноді називають словником. Він містить константи, змінні і оператори.

Множина Q може бути скінченною або нескінченною. У другому випадку вона, як правило, задається за допомогою скінченної множини схем аксіом і правил породження окремих аксіом зі схеми аксіом. Зазвичай аксіоми поділяються на два види: логічні аксіоми (загальні для цілого класу формальних теорій) і нелогічні (або власні) аксіоми, які визначають специфіку і зміст окремої теорії.

2.2 Правила виведення у формальних системах

У формальних системах формальне доведення визначається як скінченна послідовність формул $M_1, M_2, ..., M_k$ така, що кожна формула M_i є або аксіомою, або виведена за допомогою одного з правил виведення, з попередніх формул M_a, де $a < i$.

Формула t називається теоремою, якщо існує доведення, в якому вона є останньою, тобто $M_r = t$. Зокрема кожна аксіома є теоремою. Те , що T є теоремою позначається як t.

Нехай $F_1, F_2, \ldots, F_n$, G є формулами ФС. Якщо існує таке правило виведення R, що $(F_1, F_2, \ldots, F_n, G) \in R$, то кажуть, що формула G є такою, що безпосередньо виводиться з формул $F_1, F_2, \ldots, F_n$ за правилом виведення R. Цей факт записують у такому вигляді:

$$\frac{F_1, F_2, \ldots, F_n}{G} \underline{R},$$

де формули $F_1, F_2, \ldots, F_n$ називаються посиланнями;

G – висновком.

Позначення правила виведення справа від риски часто опускають, якщо воно зрозуміле з контексту.

Якщо у ФС існує виведення формули G з формул $F_1, F_2, \ldots, F_n$, то це записують у такому вигляді:

$$F_1, F_2, \ldots, F_n \vdash G,$$

де формули $F_1, F_2, \ldots, F_n$ називають гіпотезами виведення.

Позначення $\vdash G$ означає, що формула G є теоремою ФС, тобто може бути виведена тільки з аксіом, без використання будь-яких гіпотез.

Якщо виконується $\Gamma \vdash G$, то $\Gamma, \Pi \vdash G$, де Г і П будь-які множини формул (тобто, якщо додавати зайві гіпотези, виведення зберігається).

Правила виведення називають також правилами словоутворення або правилами виведення висновку. В принципі вони дозволяють визначити, чи є певна формула теоремою даної ФС.

Розрізняють два типи правил виведення:

1) правила, які застосовують до формул, що розглядаються як єдине ціле (такі правила називають продукціями);

2) правила, які можна застосовувати до будь-якої окремої частини формули, причому самі ці частини є формулами, що входять до ФС. Такі правила називають правилами переписування.

Наприклад, в математиці правило виведення

$$X < Y \text{ і } Y < Z \text{ обумовлює } X < Z$$

застосовується до формули як до єдиного цілого – це є продукція. В продукціях слово "обумовлює" (спричиняє, викликає) замінюють символом $\rightarrow$.

На відміну від попереднього, правило

$$X - X \vdash 0$$

має сенс для будь-якого вигляду виразу, що входить в нього як X. Це є правило переписування і в цьому випадку слово "обумовлює" замінюється спеціальним символом $\mapsto$. І продукція, і переписування мають тільки один напрямок виведення – справа – наліво.

У будь-яких формальних системах використовуються також правила підстановки. Вони дуже часто бувають необхідними тому, що для застосування правила виведення до будь-якої формули М з певної формальної системи, необхідно зіставити лівий елемент правила Q з М. Це й здійснюється за допомогою підстановок в Q і/або в М.

Підстановка полягає у заміні усіх складових деякої змінної на формулу з ФС, що не містить цієї змінної.

2.3 Інтерпретація та розв'язуваність формальних систем

Формальні системи (ФС) є не просто грою розуму, а завжди є моделлю певної реальності (конкретної або математичної).

Інтерпретація подає собою розповсюдження вихідних положень певної формальної системи на реальний світ. Вона дає зміст кожному символу ФС і встановлює взаємно однозначну відповідність між символами формальної системи і реальними об'єктами. Теореми ФС після інтерпретації стають твердженнями у звичайному сенсі слова, після чого вже можна робити висновки про їх істинність або хибність.

Спочатку вчений вивчає реальність, конструюючи деяке абстрактне уявлення про неї, тобто деяку формальну систему. Потім він доводить теореми формальної системи. І, кінець-кінцем, дає інтерпретацію теорем, отриманих під час формалізації. Ясна річ, для кожної ФС завжди має існувати хоча б одна інтерпретація, в якій кожна теорема даної ФС буде істинною.

Зауважимо, що одна й та ж сама формальна система може служити моделлю великої кількості різноманітних конкретних ситуацій.

Крім того, ФС є тим більш цікавою, чим більше існує для неї різних інтерпретацій. При цьому наявність навіть одного формального доведення забезпечує отримання великої кількості конкретних результатів.

Отже, інтерпретацією *I* формальної системи ФС в область інтерпретації М називається функція:

$$I\colon \text{ФС} \rightarrow \text{М},$$

яка кожній формулі формальної системи ФС однозначно зіставляє деяке змістове твердження відносно об'єктів множини (алгебраїчної системи) М. Це твердження може бути істинним або хибним (або взагалі не мати істинністного значення). Якщо відповідне твердження є істинним, кажуть, що формула виконується в даній інтерпретації.

Множина Г істинних формул ФС називається теорією даної предметної області, а кожна окрема формула – аксіомою. Якщо теорія дійсно адекватно описує предметну область, то всі факти про неї, які є істинними, будуть випливати з аксіом даної теорії, і жоден хибний факт не буде випливати з них. Теорія називається повною (або адекватною), якщо всі істинні факти заданої предметної області випливають з неї (кожному істинному твердженню М відповідає теорема теорії).

Інтерпретація, в якій формула стає істинною, називається моделлю даної формули.

Інтерпретація *I* називається моделлю множини формул Г, якщо всі формули цієї множини виконуються в інтерпретації.

Інтерпретація *I* називається моделлю теорії, якщо всі аксіоми цієї теорії є істинними в інтерпретації I.

Інтерпретація *I* називається моделлю формальної системи ФС, якщо всі теореми цієї ФС виконуються в інтерпретації *I* (тобто всі формули, які можна вивести в даній ФС, виявляються істинними в даній інтерпретації).

Формули ФС мають такі основні семантичні властивості:

- загальнозначимість – якщо не існує жодної інтерпретації, в якій формула приймає значення «хибність»;
- суперечність – якщо не існує жодної інтерпретації, в якій формула приймає значення «істина»;
- здійсненність – якщо існує хоча б одна інтерпретація, в якій формула приймає значення «істина».

Формула G називається логічним висновком множини формул Г, якщо G виконується у будь-якій моделі Г.

Теорія називається семантично несуперечною, якщо жодна її теорема не є суперечною. Модель формальної теорії існує тоді і тільки тоді, коли ця теорія є семантично несуперечною.

Теорія називається формально несуперечною, якщо в ній не можна водночас довести істинність формул F і $\neg$F.

Система аксіом формально несуперечної теорії називається незалежною, якщо жодна з аксіом не може бути виведена з інших відповідно до правил виведення ФС.

Формальна система називається розв'язуваною, якщо існує алгоритм, який для будь-якої формули ФС визначає, чи є ця формула теоремою даної формальної системи. ФС називається напіврозв'язуваною, якщо існує алгоритм, який для будь-якої її формули видає позитивну відповідь, якщо формула є теоремою ФС і, можливо, не видає ніякої відповіді, якщо формула не є теоремою (тобто, алгоритм можна застосовувати не до всіх формул). Причина можливої відсутності алгоритму розв'язання ФС полягає у тому, що навіть якщо застосовувати правила словоутворення послідовно до всіх можливих об'єктів ФС і ФС принципово дозволяє здійснити повний перебір її теорем (навіть для їх нескінченної кількості), все ж таки в загальному випадку не існує ніякого способу, для перебору всіх її нетеорем.

2.4 Контрольні запитання

1. Що таке формальна система?
2. Наведіть означення поняття „теорема".
3. Які основні типи правил виведення Ви знаєте?
4. Що таке „інтерпретація"?
5. Чи може формальна система:
   а) не мати жодної інтерпретації;
   б) мати лише одну інтерпретацію;
   в) мати велику кількість різноманітних інтерпретацій?
6. Що означає термін „теорія предметної області"?
7. Що є моделлю теорії?
8. Назвіть основні семантичні властивості формул формальної системи.
9. Яка формальна система називається розв'язуваною?

3 ЛОГІКА ВИСЛОВЛЕНЬ

3.1 Поняття висловлення та висловлювальної форми

Як ми вже відмічали, відношенням називають відображення n елементів з області інтерпретації на один з двох елементів множини {істина, хибність}. Найбільш звичними для нас є арифметичні відношення, наприклад, такі як "більше" або "менше".

Але на множину {істина, хибність} можна відобразити і значну кількість речень природної мови. Такі речення називають висловленнями. Наприклад, висловлення

"Київ – столиця України"

є істинним, а висловлення

"Волга впадає в Чорне море"

хибним.

Оскільки з точки зору логіки все, що цікавить нас у цих реченнях, є їхні значення істинності, формалізація цих речень зводиться до позначення їх відповідними літерами, наприклад, A або B, які будуть подавати змінні, що можуть приймати лише одно з двох значень: I (1) – істина, або II (0) – хибність.

Не всяке речення є висловленням. Наприклад, до висловлень не можна віднести питальні та окличні речення, оскільки вести мову про їх істинність не має сенсу. Дійсно, що можна сказати про істинність таких речень, як

"Стій!" та "Хто іде?"

Не є висловленнями і такі речення як

"Галя – дуже красива дівчина",
"Логічне програмування – цікавий предмет",
"Манна каша смачна",

оскільки немає і об'єктивно не може бути єдиної думки з цього приводу. Такі речення відображають суб'єктивну думку особи і тому можуть мати різні значення істинності для різних суб'єктів.

У той же час, речення:

"Існують позаземні цивілізації"
"Існує десять невідомих нині поховань фараонів"

слід вважати висловленнями. Хоча сьогодні ніхто не знає точного значення істинності цих висловлень, але об'єктивно вони є або істинними, або хибними.

Не є висловленнями і такі речення як

"Йшов сніг",
"Площа аудиторії дорівнює 20 м^2",
"$A^2=4$",

оскільки для встановлення їх істинності потрібні додаткові дані, а саме: коли і де йшов сніг, про яку аудиторію йдеться, яке саме число позначено буквою A.

В останньому прикладі A може позначати не конкретне число, а бути змінною. В цьому випадку кожному конкретному значенню змінної буде відповідати або істинне, або хибне висловлення, наприклад $2^2=4$ – істинне висловлення, а $2^3=4$ – хибне.

Речення, які містять хоча б одну змінну і стають висловленнями після підстановки замість всіх змінних їх конкретних значень, називають висловлювальними формами. Наприклад, речення

"Він рудий",
"Число ділиться на 7"

не містять змінних в явному вигляді, але вони є все ж таки висловлювальними формами, оскільки стають висловленнями тільки після підстановки в них, відповідно, чоловічого імені та конкретного числового значення. Тобто, ці речення можна було б записати у іншому вигляді:

"Людина X має руде волосся",
"Число Y ділиться на 7".

Зауважимо, що з висловлювальних форм можна отримати висловлення не тільки шляхом підстановок на місця змінних конкретних значень, але й за допомогою використання таких спеціальних слів, як

"всякий" (або будь-який, чи кожний)

та

"існує" (або деякий, хоча б один).

Наприклад, висловлення "Всяке число ділиться на 7" є хибним, а висловлення "Існує число, яке ділиться на 7" – є істинним.

3.2 Логічні зв'язки

Висловлення, як і речення, можуть бути елементарними або складеними.

Елементарним висловленням будемо називати таке висловлення, яке неможливо розділити на декілька більш дрібних висловлень. Наприклад, речення

"У групі 1ІС навчається студент Бойко, сестра якого навчається в цій же групі"

можна подати одним висловленням, а можна й двома:

"У групі 1ІС навчається студент Бойко" і
"Сестра студента Бойко навчається у тій же групі".

Вважати деяке висловлення за елементарне чи не вважати, в кожному окремому випадку залежить від нашої волі. Це не має принципового значення у межах тієї системи логічного виведення, яка використовується у логіці висловлень.

Але з декількох елементарних висловлень, що є простими реченнями, можна будувати складені висловлення, які є складеними реченнями природної мови. Це здійснюється з використанням спеціальних логічних зв'язок, що відповідають сполучникам "і", "а", "або", "якщо...то", "тоді і тільки тоді" та ін. Нове речення можна отримати і за допомогою частки "не".

Наприклад, з елементарних висловлень

"Я поїду до Києва" і "Ти поїдеш до Львова"

можна створити такі складені висловлення:

"Я поїду до Києва і ти поїдеш до Львова",
"Я поїду до Києва або ти поїдеш до Львова",
"Якщо я поїду до Києва, то ти поїдеш до Львова",
"Я не поїду до Києва"

та ін.

Зауважимо, що елементарне, з точки зору логіки, висловлення не рівнозначне простому реченню з точки зору граматики. Наприклад просте речення "Завтра не буде дощу" не є простим за своєю логічною структурою. Воно створене з елементарного речення "Завтра буде дощ" за допомогою логічної зв'язки "не".

Речення, які побудовані за допомогою зв'язки "не" мають назву заперечень, за допомогою зв'язки "і"– диз'юнкцій, за допомогою зв'язки "або" – кон'юнкцій. Значення істинності складених речень визначаються з урахуванням значень істинності елементарних речень за допомогою таблиці істинності.

Таблиця 1 – Таблиця істинності логічних зв'язок

Елементарні речення		Логічні зв'язки				
A	B	І	АБО	Імплікація	Еквіваленція	НІ
		$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	$\neg$
0	0	0	0	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	0	0
1	1	1	1	1	1	0

Між знаками логічних зв'язок і конструкціями природної мови є певний зв'язок, який проілюструємо на прикладах.

Якщо, наприклад, буквою А позначено речення

А = "Завтра буде тепла погода",

а буквою В – речення

В = "Завтра не буде дощу",

то виразу $A \wedge B$ буде відповідати речення

$A \wedge B$ = "Завтра буде тепла погода і не буде дощу".

Якщо ж назавтра буде холодно або піде дощ, то прогноз потрібно буде визнати хибним. Істинним він буде лише в тому випадку, якщо істинними виявляться і речення А, і речення В, що відповідає наведеній таблиці істинності логічних зв'язок (табл. 1).

Особливістю сполучника "або" є те, що він може мати або розділювальний, або нерозділювальний зміст, від чого може залежати і істинність складеного речення.

Наприклад, розглянемо речення

"Число 2 просте, або число 2 – парне".

Воно складається з двох істинних висловлень

А = "Число 2 просте"

та

В = "Число 2 парне".

Чи можемо ми подати його у вигляді диз'юнкції:

$A \vee B$ = "Число 2 просте, або число 2 – парне"?

Якщо уважно придивитися до останньої формули, можна побачити у ній певну некоректність. Справа в тому, що вираз $A \vee B$, згідно з таблицею істинності, має бути істинним не тільки у тих випадках, коли істинними є складові А або В виразу, але й тоді, коли А і В є істинними одночасно. Але подібний випадок у нашому прикладі неможливий. Число може бути або простим, або парним. Але водночас бути і простим, і парним воно не може. Тобто, одночасна істинність А і В мала б сигналізувати про хибність відповідного твердження, але диз'юнкція стверджує, що воно істинне.

Таку ситуацію можна виправити введенням спеціальної зв'язки, що отримала назву розділювальної диз'юнкції. Але ми цього робити не будемо, оскільки таку зв'язку можна подати комбінацією вже відомих нам логічних зв'язок:

$$(\neg A \wedge B) \vee (A \wedge \neg B).$$

Перевіримо, чи буде досягнута потрібна мета з використанням такої формули? Якщо А і В будуть хибними, то хибними будуть кон'юнкції $\neg A \wedge B$ і $A \wedge \neg B$. Отже, згідно з таблицею істинності диз'юнкції, хибною буде і вся формула. Якщо ж А і В будуть водночас істинними, то знов обидві кон'юнкції стануть хибними, оскільки до них входять заперечення висловлень А і В. Отже, знов вся диз'юнкція стане хибною. І лише коли одне з висловлень А або В буде істинним, а інше – хибним, істинною стане й уся формула.

Отже, якщо сполучник "або" є розділювальним (як, наприклад, і в реченні "дві прямі на площині паралельні або перетинаються"), то отримане за допомогою диз'юнкції складене речення буде хибним. Якщо ж сполучник не є розділювальним, як це і мають на увазі, коли його використовують як логічну зв'язку, то складене речення буде істинним.

Однією з найцікавіших логічних зв'язок є імплікація, яка відповідає сполучникові "якщо...то...". Вона позначається символом $\rightarrow$, при цьому запис $A \rightarrow B$, читається так:

"Якщо А, то В", або "А імплікує В".

Імплікація відповідає змісту сполучника "якщо... то...", наприклад, в реченні

"Якщо число А ділиться на 4, то воно ділиться і на 2".

В той же час, вона очевидно, не відповідає змісту сполучника "якщо... то..." в реченні

"Якщо Бойко захоплюється логічним програмуванням, то Луцюк нічим окрім футболу не цікавиться".

Тобто, імплікація відображає деякий причинно-наслідковий зв'язок, який існує у навколишньому середовищі.

Ліва частина імплікації, що описує причину, називається антецедентом, а права, що описує висновок, – консеквентом. З означення імплікації згідно з таблицею істинності випливає, що імплікація з хибним антецедентом або істинним консеквентом завжди істинна, і що імплікація буває хибною тільки в тому випадку, якщо її антецедент істинний, а консеквент – хибний.

Особливості таблиці істинності імплікації випливають з того, що вона відображає один єдиний причинно-наслідковий зв'язок, що існує у навколишньому середовищі, нічого не кажучи про наявність або відсутність будь-яких інших причин, що можуть викликати той самий висновок, що і причина, яка розглядається (рис. 3).

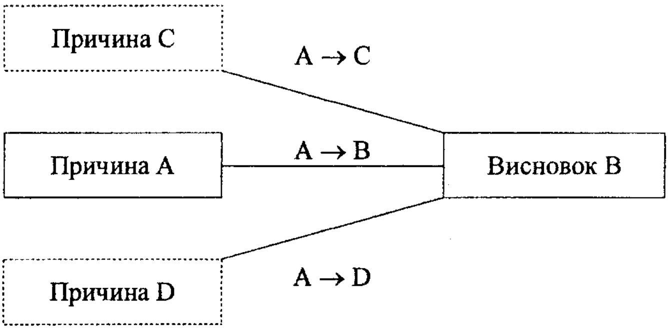


Рисунок 3 – Пояснення до таблиці істинності імплікації

Твердження про існування причинно-наслідкового зв'язку між A і B означає гарантію того, що спостерігаючи причину A обов'язково спостерігається і висновок B. Тому четвертий рядок таблиці істинності імплікації містить значення 1, яке означає, що ситуація одночасного спостерігання A і B відповідає наявності між ними причинно-наслідкового зв'язку.

У той же час, твердження про існування причинно-наслідкового зв'язку між причиною A та висновком B виключає можливість існування будь-якої ситуації, в якій антецедент імплікації є істинним, а консеквент хибним. Якщо б така ситуація існувала, це свідчило б про можливість одночасної наявності причини і відсутності висновку, що заперечує існування причинно-наслідкового зв'язку. Тому третій рядок таблиці істинності імплікації містить значення 0, яке відображає хибність даної ситуації.

Розглядаючи далі логічні зв'язки імплікації ще раз підкреслимо, що її застосування відображає лише твердження про існування одного, окремо взятого причинно-наслідкового зв'язку між причиною A і висновком B, і нічого не каже про можливу наявність або відсутність інших причин, які можуть викликати висновок B. Отже, відсутність причини A зовсім не гарантує відсутність висновку B, оскільки він може бути викликаний іншою причиною (C і D на рис. 3), про наявність або відсутність яких у імплікації $A \rightarrow B$ будь-яка інформація відсутня. Тому перший і другий рядки таблиці істинності імплікації містять значення 1, яке означає, що наявність або відсутність висновку B в умовах відсутності причини A не заперечує твердженню про існування між ними причинно-наслідкового зв'язку.

Розглянемо приклад застосування імплікації в математиці, під час описання одного з причинно-наслідкових зв'язків, що існує у ній. Розглянемо твердження про наявність для будь-якого натурального *a* такого причинно-наслідкового зв'язку:

"Якщо число А ділиться на 4, то воно ділиться і на 2"

Здійснимо формалізацію цього твердження:

A = "Число А ділиться на 4"
B = "Число А ділиться на 2"
$A \to B$

Визнаючи істинним існування даного причинно-наслідкового зв'язку, ми повинні визнати істинними і такі висловлення як:

"Якщо 16 ділиться на 4, то воно ділиться і на 2",
"Якщо 18 ділиться на 4, то воно ділиться і на 2",
"Якщо 17 ділиться на 4, то воно ділиться і на 2",

що відповідають четвертому, другому та першому рядкам таблиці істинності імплікації (табл. 2), відповідно:

Таблиця 2 – Таблиця істинності імплікації

Номер	A	B	$A \to B$
1	0	0	1
2	0	1	1
3	1	0	0
4	1	1	1

Хибного висловлення, яке б відповідало другому рядку таблиці істинності (табл. 2), в даному випадку отримати неможливо в силу справедливості вихідного твердження.

Розглянемо інший, нематематичний, приклад, в якому хлопець обіцяє дівчині:

"Якщо буде хороша погода, то я прийду до тебе в гості".

Якщо погода буде хорошою, то друг повинен прийти в гості і треба підготуватися до його зустрічі. Але, що буде, якщо погода буде поганою? Для імплікації хибність антецедента означає, що в цьому випадку друг може прийти в гості, а може й ні. Складене висловлення в обох випадках буде істинним і невідомо, чи треба дівчині готуватися до зустрічі друга, виділяючи на це час, готуючи смачні страви, здійснюючи вибір нарядів та макіяжу, чи не треба.

Якщо ж висунути умову, щоб при хибності антецедента було б хибним й усе складене твердження цілком, то це відповідало б поверненню від операції імплікації до операції кон'юнкції.

Зауважимо також, що згідно з означенням імплікації, істинними будуть і такі речення, як, наприклад,

"Якщо $2 \cdot 2=4$, то Київ столиця України",
"Якщо місяць зроблений із зеленого сиру, то коні літають".

Такі речення здаються нісенітницею, але тільки тому, що ми звикли зв'язувати сполучником "якщо... то..." речення, зв'язані за змістом. З формальної ж точки зору, під час означення логічних операцій зміст речень ніяк не враховується, вони розглядаються лише як об'єкти, що мають єдину властивість, бути істинними або хибними.

Логічна операція, яка відповідає сполучникові "тоді і тільки тоді, коли" має назву еквіваленції і позначається символом "$\leftrightarrow$".

Наприклад, кажучи:

"Я поїду до Києва *тоді і тільки тоді, коли* ти поїдеш до Львова"

ми стверджуємо, що або водночас станеться і те, і інше, або не станеться ні того, ні іншого. Тобто, висловлення вигляду $A \leftrightarrow B$ буде істинним тільки якщо значення істинності висловлень A і B будуть збігатися, і хибним в іншому випадку.

3.3 Формули логіки висловлень. Поняття тавтології

Логіка висловлень є одним з основних і в той же час найпростіших розділів математичної логіки. Елементарні висловлення розглядаються в ній як "нероздільні" атоми, а складені висловлення, як молекули, створені з атомів шляхом застосування логічних операцій. Якщо єдиною важливою властивістю елементарних висловлень є значення їх істинності, то в складених висловленнях інтерес викликає, в першу чергу, їх структура, яка відображає спосіб, яким їх було створено. Структура складених висловлень визначає залежність їх значень істинності від значень істинності складових елементарних висловлень.

Нехай X, Y, Z – змінні, замість яких можна підставляти елементарні висловлення. Такі змінні називають висловленими змінними. За допомогою таких змінних і символів логічних операцій будь–яке висловлення можна формалізувати, тобто замінити формулою, яка відображає його структуру. Наприклад, висловлення

"Якщо 100 ділиться на 2 і на 5, то воно ділиться і на 10"

можна формалізувати у вигляді $(X \wedge Y) \rightarrow Z$, де

$$X = \text{“100 ділиться на 2”},$$
$$Y = \text{“100 ділиться на 5”},$$
$$Z = \text{“100 ділиться на 10”}.$$

Задамо алфавіт логіки висловлень, тобто перелік усіх символів, які дозволяється використовувати під час описання формул.

1. X, Y, Z, X_i, Y_i, Z_i (i – натуральне число) – символи для позначення висловлювальних змінних.
2. I, II – символи для позначення логічних констант «істина» та «хибність». Символ II тут вибраний для більшої зручності описання, оскільки символ X зазвичай використовується для описання змінних і використання його для описання константи викличе значні незручності.
3. $\wedge, \vee, \rightarrow, \leftrightarrow, \neg$ – символи логічних операцій.
4. (,) – дужки та кома (допоміжні символи для означення порядку виконання операцій).

Тепер можна дати чітке означення формули логіки висловлень.

1. Будь-яка висловлена змінна є формулою.
2. Символи I, II є формулами.
3. Якщо F формула, то $\neg F$ теж формула.
4. Якщо F_1 і F_2 формули, то $(F_1 \wedge F_2)$, $(F_1 \vee F_2)$,$(F_1 \rightarrow F_2)$,$(F_1 \leftrightarrow F_2)$ теж формули.
5. Будь-які інші формули в логіці висловлень відсутні.

Серед всієї множини формул логіки висловлень можна виділити такі, які приймають значення I на будь-яких наборах значень змінних, що входять до їх складу. Такі формули, як і формулу I – називають тавтологіями або тотожно істинними формулами. Встановлення істинності речення виконують, як правило, шляхом перевірки відповідності його змісту дійсності. Наприклад, твердження

«А. С. Пушкін народився у 1799 році»

є істинним, оскільки воно відповідає дійсності. В той же час твердження

«10 липня 2003 р. буде понеділком»

не відповідає дійсності і тому буде хибним.

Однак, для встановлення істинності, наприклад, речення

«Дві прямі на площині паралельні або перетинаються» не треба з'ясовувати про які саме прямі йдеться. Зрозуміло, що це речення свідомо істинне. Важливість тавтологій в логіці полягає в тому, що вони відображають логічну структуру речень, істинність яких обумовлюється тільки саме їх структурою. Через поняття тавтології можна визначити також ряд інших важливих понять, зокрема, таких як рівнозначність та логічний висновок.

3.4 Контрольні запитання

1. Що таке висловлення та висловлена форма?
2. Наведіть власні приклади речень, які не є висловленнями. Поясніть, чому саме вони не є висловленнями.
3. Які шляхи перетворення висловлювальних формул на висловлення Ви знаєте?
4. Наведіть власні приклади речень, які містять розділювальне і нерозділювальне АБО. Поясніть різницю між цими видами логічної зв'язки АБО.
5. Назвіть відомі Вам з булевої алгебри логічні зв'язки і наведіть їх таблиці істинності.
6. Які логічні зв'язки використовуються в логіці висловлень?
7. Наявність якого відношення між об'єктами відображає логічна зв'язка імплікації?
8. Поясніть фізичний зміст таблиці істинності імплікації.
9. Який мінімальний набір логічних зв'язок необхідний для формалізації будь-якого висловлення? Наведіть приклади подання усіх інших відомих Вам логічних зв'язок за допомогою зв'язок з вибраного мінімального набору.
10. Вкажіть які з наведених речень є висловленнями, а які ні і чому:

    а) Сонце – супутник Землі;

    б) 2+3=4?;

    в) сьогодні чудова погода;

    г) у романі Л. Н. Толстого „Війна і мир" 3 432 536 слів;

    д) Київ розташований на Дніпрі;

    е) музика Баха занадто складна;

    ж) перша космічна швидкість дорівнює 7,8 км/сек.;

    з) залізо – метал;

    и) якщо один кут у трикутнику прямий, то трикутник буде тупокутним.
11. Сформулюйте заперечення нижченаведених висловлень або висловлювальних форм:

    а) в мішень влучили з першого пострілу;

    б) цей ранок ясний і теплий;

    в) число n ділиться на 2 або на 3;

г) цей трикутник рівнобедрений і прямокутний;

д) на контрольній роботі кожен учень писав своєю ручкою.

12. Визначте, які з наведених речень є логічно істинними:

а) трикутник АВС прямокутний або гострокутний;

б) якщо Петро здоровий, то він здоровий і щасливий;

в) якщо Микола здоровий, то він здоровий або щасливий;

г) якщо A<>0 або C<>0, то неправильно, що A=0 або C=0.

13. На запитання „Хто з трьох студентів вивчав математичну логіку?" отримано правдиву відповідь: „Якщо вивчав Іван, то вивчала і Галя, але неправильно, що якщо вивчала Галя, то вивчав і Ігор". Формалізуйте відповідь і з побудованої формули визначте, хто ж саме вивчав математичну логіку.

14. Формалізуйте такий висновок: „Якщо А і В істинні, то С – істинне. Але С – хибне: Отже, А або В хибні".

4 ЛОГІЧНИЙ ВИСНОВОК У ЛОГІЦІ ВИСЛОВЛЕНЬ

4.1 Рівнозначність логічних формул

Формули F_1 і F_2 називають рівнозначними, якщо їх еквіваленція $F_1 \leftrightarrow F_2$ є тавтологією, тобто, якщо вони одночасно приймають значення І та Х. Приклади рівнозначних речень можуть бути різноманітними:

«Завтра буде вівторок»
«Вчора була неділя»

у понеділок такі висловлення істинні, а в інші дні тижня – хибні;

«X=2»
«2X=4»

приймають однакові значення істинності якщо однакові значення змінних;

«Завтра буде дощ»
«Неправильно, що завтра не буде дощу»

одночасно підтвердяться або ні.

Кажуть, що висловлення P_1 та P_2 рівнозначні в логіці висловлень, якщо рівнозначними є відповідні їм формули.

Рівнозначність формул логіки висловлень часто називають законами логіки, найбільш важливими з яких вважаються такі:

1) $X=X$ – закон тотожності: думка, що є змістом деякого висловлення, залишається незмінною протягом всіх міркувань, в яких використовується це висловлення;

2) $X \wedge \neg X = Х$ – закон суперечності: жоден розв'язок не може бути істинним водночас із своїм запереченням;

3) $X \vee \neg X = I$ – закон виключення третього: висловлення може бути або істинним, або хибним, іншого не дано;

4) $\neg(\neg X)=X$ – закон подвійного заперечення: заперечувати заперечення деякого висловлення – те ж саме, що його стверджувати;

5) $X \wedge X=X; X \vee X=X$ – закони ідемпотентності;

6) $X \wedge Y=Y \wedge X; X \vee Y=Y \vee X$ – закон комутативності;

7) $(X \wedge Y) \wedge Z=X \wedge (Y \wedge Z)$; $(X \vee Y) \vee Z=X \vee (Y \vee Z)$ – закони асоціативності (відмітимо, що закони комутативності та асоціативності аналогічні до законів множення та додавання);

8) $X \wedge (Y \vee Z)=(X \wedge Y) \vee (X \vee Z); X \vee (Y \wedge Z)=(X \vee Y) \wedge (X \vee Z)$ – закони дистрибутивності (зауважимо, що на відміну від додавання та множення чисел, ло-

гічні додавання та множення рівноправні відносно закону дистрибутивності);

9) $\neg(X\wedge Y)=\neg X\vee\neg Y$; $\neg(X\vee Y)=\neg X\wedge\neg Y$ – закони де Моргана (заперечення кон'юнкції рівнозначне диз'юнкції заперечень і навпаки).

Наведені закони логіки використовуються для перетворення формул з метою їх спрощення, головними критеріями якого є відсутність заперечення неелементарних формул та зменшення кількості знаків кон'юнкції та диз'юнкції. З цією метою також часто використовують і такі рівнозначності:

10) $X\wedge I=X; X\vee II=X$;

11) $X\wedge II=II; X\vee I=I$;

12) $X\wedge(X\vee Y)=X; X\vee(X\wedge Y)=X$ – закони поглинання;

13) $(X\vee Y)\wedge(\neg X\vee Y)=Y$; $(X\wedge Y)\vee(\neg X\wedge Y)=Y$ – закони склеювання.

Підкреслимо, що в формулах 5–13 між парами рівнозначностей проявляється так званий принцип двоїстості: дві формули, які не містять символів $\to$ та $\leftrightarrow$ називають двоїстими, якщо одну з них можна отримати з іншої шляхом заміни всіх символів $\wedge$, $\vee$, I, II на символи $\vee$, $\wedge$, II, I, відповідно. Принцип двоїстості стверджує, що якщо дві формули, які не містять символів $\to$ та $\leftrightarrow$ рівнозначні, то рівнозначні і двоїсті до них формули. При цьому слід враховувати, що знаки $\to$ та $\leftrightarrow$ завжди можна виразити через кон'юнкцію та заперечення за допомогою таких формул:

$$X \to Y = \neg X \vee Y;$$
$$X \leftrightarrow Y = (X \to Y)(Y \to X) = (\neg X \vee Y)(\neg Y \vee X).$$

Зауважимо, що для кожного речення $A \to B$, можна скласти три відповідних речення:

1. $B \to A$ – обернене;
2. $\neg A \to \neg B$ – протилежне;
3. $\neg B \to \neg A$ – обернено-протилежне.

В логіці існує **закон контрапозиції**, який стверджує таку рівнозначність:

$$X \to Y = \neg Y \to \neg X.$$

Тобто, якщо X є теоремою (реченням яке можна довести), то теоремою є і $\neg Y \to \neg X$. Наприклад, нехай треба довести твердження

«Якщо C^2 непарне, то і C непарне».

Сформулюємо і доведемо обернено-протилежну теорему:

«Якщо с парне, то і C^2 парне».

Дійсно, якщо с парне, то $C=2P$, $C^2=4P^2=2(2P^2)=2Y$, тобто C^2 – теж парне. Таким чином обернено-протилежне даному речення доведено, тобто доведено і вихідне речення.

Якщо в формулі $X \to Y = \neg Y \to \neg X$ поміняти місцями X і Y, то отримаємо: $Y \to X = \neg X \to \neg Y$. Звідси випливає, що речення протилежне і обернене даному – рівнозначне, тобто достатньо довести або спростувати одне з них, щоб довести або спростувати й інше.

Якщо речення $A \to B$ є теоремою, то кажуть, що A є достатньою умовою B, а B – необхідною умовою A. Дійсно, якщо імплікація $A \to B$ тотожно істинна, то відомостей про істинність A достатньо для того, щоб стверджувати, що істинним є і B (інакше імплікація стала би хибною, тобто не була б тотожно істинною). В той же час, для того щоб A могло стати істинним, необхідно, щоб істинним було B (у тотожно істинній імплікації $A \to B$, A може приймати істинне значення лише при істинному B, а при хибному B – не може).

Якщо ж теоремами є обидва взаємно-зворотних речення $A \to B$ і $B \to A$, то і речення $A \leftrightarrow B$ є теоремою $(A \leftrightarrow B) = (A \to B)(B \to A)$, внаслідок чого A стає необхідною і достатньою умовою для B і навпаки.

Дійсно, B може стати істинним у тотожно істинній еквіваленції тільки за умови істинності A (дивіться таблицю істинності еквіваленції на рис. 3). Але істинність A у цьому випадку гарантує і істинність B (тобто, є достатньою умовою для істинності B).

Якщо ж $A \to B$ є теоремою, а $B \to A$ не є теоремою, то A є достатньою але не необхідною умовою B, а B – необхідною, але недостатньою умовою A.

Дійсно, з тотожної істинності імплікації $A \to B$ випливає, що наявність відомостей про істинність A є достатньою для впевненості у істинності B. У той же час, згідно з таблицею істинності імплікації (рис. 4), B може бути істинним і при хибному A, про що, у даному випадку, свідчить дозволеність хибного значення імплікації $B \to A$.

4.2 Відношення логічного висновку

Коли кажуть, що одне речення P2 випливає з іншого речення P1, то мають на увазі, що кожен раз, коли істинним є речення P1, істинним буде і речення P2. В логіці висловлень це означає, що для формул F1 і F2, які відповідають реченням P1 і P2, відсутній такий набір змінних, в якому F1 стала б істинною, а F2 – хибною. Тобто, формула F2 випливає з формули F1, якщо імплікація $F1 \to F2$ є тавтологією.

Назвемо аргументом сукупність речень, про одне з яких, що має назву висновку, кажуть, що воно випливає з інших, які називають посиланнями (посилання може бути і одне). Аргумент називають правильним, якщо з кон'юнкції його посилань випливає висновок. Тобто, для встановлення правильності аргумента треба:

а) формалізувати всі посилання та висновок;

б) скласти кон'юнкцію формалізованих посилань;

в) перевірити, чи випливає з отриманої формули така формула, яка відповідає висновку. Якщо це так – аргумент правильний, якщо інакше – неправильний.

Під час запису аргументів над горизонтальною рискою записують посилання, а під нею – висновок. Наприклад, розглянемо такий аргумент:

«Якщо чотирикутник ABCD це ромб, то його діагоналі взаємно перпендикулярні. Чотирикутник ABCD – ромб. Отже, його діагоналі взаємно перпендикулярні».

Запишемо його в формалізованому вигляді, для чого спочатку формалізуємо наведені твердження:

X = "Чотирикутник ABCD – ромб";

Y = "Діагоналі чотирикутника ABCD – взаємно перпендикулярні".

З урахуванням наведеної формалізації аргумент набуде такого вигляду:

$$\begin{array}{|l} X \to Y \\ X \\ \hline Y \end{array}$$

Легко впевнитися, що цей аргумент правильний. Дійсно, формула

$$[(X \to Y) \land X] \to Y,$$

що відповідає даному аргументу, є тавтологією.

Для перевірки істинності наведеної формули можна побудувати її таблицю істинності, яка наведена на таблиці 3. Дані таблиці свідчать, що формула буде істинною за будь-яких комбінацій значень змінних X і Y.

Таблиця 3 – Таблиця істинності для перевірки правильності аргументу

X	Y	$X \to Y$	$(X \to Y) \land X$	$[(X \to Y) \land X] \to Y$
0	0	1	0	1
0	1	1	0	1
1	0	0	0	1
1	1	1	1	1

Але на практиці для доведення тавтології імплікації зовсім не обов'язково будувати її повну таблицю істинності. При цьому достатньо лише впевнитися у відсутності такої комбінації значень змінних, що входять до складу імплікації, за яких її антецедент стане істинним, а консеквент – хибним.

Для нашого прикладу це означає, необхідність доведення неможливості одночасного виконання таких умов:

антецедент: $[(X \to Y) \wedge X] = 1$;

консеквент: $Y=0$.

Підставивши значення Y до антецедента, отримаємо:

$$[(X \to 0) \wedge X] = 1.$$

Очевидно, що остання формула є тотожно хибною, оскільки кон'юнкція у її лівій частині ніколи не може бути істинною. Якщо $X=0$, хибність кон'юнкції обумовлюється хибністю X, а при $X=1$ – хибною стає імплікація $X \to 0$.

Отже, правильність аргументу доведено без побудови достатньо складної таблиці істинності.

Зауважимо, що в даному прикладі істинним посиланням відповідає істинний висновок.

Розглянемо інший аргумент:

«Якщо 10 ділиться на 3, то й 100 ділиться на 3.
10 ділиться на 3. Отже й 100 ділиться на 3».

В формалізованому вигляді такий аргумент буде правильним, тому що він має точно такий же вигляд, як і попередній аргумент. Однак його висновок є хибним, тому що хибними є посилання.

Розглянемо ще один приклад аргументу:

«Якщо 10 парне число, то й 100 парне число.
100 – парне число. Отже й 10 парне число»

Даному аргументу буде відповідати така формула:

$$[(X \to Y) \wedge Y] \to X.$$

Ця формула не є тавтологією, тобто з кон'юнкції посилань в даному випадку не випливає висновок. Дійсно, для значень змінних X=0 і Y=1, отримаємо ситуацію, у якій антецедент набуде істинного значення, а консеквент – хибного.

Отже, такий аргумент не є правильним, хоча висновок і є істинним. Тобто, істинність висновку зовсім не свідчить про правильність аргументу.

Таким чином, суть розбіжностей між правильним і неправильним аргументами полягає в тому, що істинність всіх посилань правильного аргументу гарантує істинність висновку. Для неправильного аргументу таке правило не виконується.

Тобто, якщо вдасться довести правильність логічного аргументу, то доведення істинності його окремих посилань вже гарантує істинність висновку без встановлення істинності складної кон'юнктивно–імплікативної формули.

Корисно запам'ятовувати такі правильні аргументи, які часто використовуються в міркуваннях:

$$\begin{array}{|l} X \to Y \\ X \\ \hline Y \end{array} \qquad\qquad \begin{array}{|l} X \to Y \\ \neg Y \\ \hline \neg X \end{array}$$

В той же час дуже поширеною логічною хибністю є використання таких неправильних аргументів:

$$\begin{array}{|l} X \to Y \\ Y \\ \hline X \end{array} \qquad\qquad \begin{array}{|l} X \to Y \\ \neg X \\ \hline \neg Y \end{array}$$

Але зовсім не обов'язково "зазубрювати" чотири наведені конструкції. Достатньо відчути логічну суть імплікації, і правильність або неправильність базових аргументів буде очевидною.

Ще раз розглянемо, чому саме перші два аргументи є правильними, а інші два – неправильними.

Нагадаємо, що імплікація $X \to Y$ відображає один окремий причинно-наслідковий зв'язок між причиною X та висновком Y. При цьому нічого не стверджується про те, що висновок Y не може мати будь-яких інших причин, тобто бути об'єктом будь-яких інших причинно-наслідкових відношень.

Припустимо, нам відомо, що деякий дід Іван завжди вирощує дуже гарний врожай картоплі. Тобто, це твердження не підлягає сумніву і є тотожно істинним (тавтологією). Подамо змінні X і Y такими твердженнями:

X = "Дід Іван посадив картоплю"

Y = "Картопля дала дуже гарний врожай"

Розглянемо тепер перший з наведених аргументів:

$$\begin{array}{|l} X \to Y \\ X \\ \hline Y \\ \hline \\ \end{array}$$

Він стверджує:

“Якщо дід Іван посадив картоплю, то картопля дасть гарний врожай”
“Дід Іван посадив картоплю”
“Отже, картопля дасть гарний врожай”

Не важко впевнитися, що між наведеними твердженнями відсутня суперечність.

Розглянемо тепер другий аргумент:

$$\begin{array}{|l} X \to Y \\ \neg Y \\ \hline \neg X \end{array}$$

Цей аргумент стверджує:

“Якщо дід Іван посадив картоплю, то картопля дасть гарний врожай”

“Картопля дала поганий врожай”
“Отже, картоплю посадив не дід Іван”

Як і в попередньому випадку, суперечність між наведеними твердженнями відсутня.

Розглянемо два неправильні аргументи:

$$\begin{array}{|l} X \to Y \\ Y \\ \hline X \end{array} \qquad\qquad \begin{array}{|l} X \to Y \\ \neg X \\ \hline \neg Y \end{array}$$

Перший з них відображає такі твердження:

“Якщо дід Іван посадив картоплю, то картопля дасть гарний врожай”
“Картопля дала гарний врожай”
“Отже, картоплю посадив дід Іван”

Такий висновок є нічим не обґрунтованим, оскільки ми нічого не стверджували про те, що крім діда Івана не існує жодної людини у світі, яка б вміла вирощувати добрі врожаї картоплі. Отже, цей аргумент є неправильним.

Те ж саме відноситься і до останнього аргументу, який стверджує:

"Якщо дід Іван посадив картоплю, то картопля дасть гарний врожай"
"Картоплю посадив не дід Іван"
"Отже, картопля дасть поганий врожай"

Цілком можливо, що існує багато людей у світі, які вирощують картоплю не гірше, а може й набагато краще, ніж дід Іван. Отже, стверджувати, що картопля дасть поганий врожай тільки тому, що її посадив не дід Іван, є логічно неправильно.

Роль розглянутих вище базових аргументів полягає в тому, що якщо звести до одного з цих наявних аргументів (за допомогою певних підстановок або перетворень), одразу можна встановити його правильність або неправильність.

4.3 Скорочений метод перевірки аргументів

Практично не завжди вдається звести наявний аргумент до одного з базових. Встановлення ж тотожної істинності імплікації, що відображає потрібний елемент, дуже часто є зовсім не тривіальним процесом (уявіть собі аргумент з кількома десятками посилань, кожна з яких відображається дуже складною логічною формулою з великою кількістю змінних). Тому існують спеціальні методи спрощеної перевірки правильності аргументів. Розглянемо один з таких методів, який отримав назву методу скороченої перевірки правильності аргументів.

Нехай задано такий аргумент:

«Якщо курс логіки неважкий, то він корисний.
Курс логіки нецікавий або цей курс некорисний.
Курс логіки цікавий.
Отже, цей курс важкий».

Проведемо формалізацію наведених тверджень:

X = «Курс логіки важкий».
Y = «Курс логіки корисний».
Z = «Курс логіки цікавий».

Побудуємо відповідний аргумент:

$$\begin{array}{|l} \neg X \to Y \\ \neg Z \lor \neg Y \\ Z \\ \hline X \end{array}$$

Припустимо тепер, що аргумент є неправильним, тобто існує такий набір X', Y', Z' значень змінних X, Y і Z, що всі формули, які відповідають посиланням мають істинні значення (I), а формула, яка відповідає висновку, має хибне значення (II).

Розглянемо висновки з такого припущення, які зведемо в таблицю, наведену на таблиці 4.

Таблиця містить чотири колонки. Перша колонка містить порядкові номери тверджень, друга і третя, відповідно, істинні і хибні твердження, четверта – коментарі відносно того, на основі яких тверджень було зроблено припущення про істинність або хибність даного твердження.

Таблиця 4 – Таблиця скороченої перевірки правильності аргументу

Номер	I	II	Коментар
1	$\neg X' \to Y'$		
2	$\neg Z' \vee \neg Y'$		
3	**Z'**		
4		X'	
5	$\neg X'$		з 4
6	Y'		з 1 і 5
7		$\neg Y'$	З 6
8	$\neg Z'$		з 2 і 7
9		**Z'**	з 8

Відповідно до нашого припущення щодо істинності всіх посилань і хибності висновку, розмістимо посилання у рядках 1-3 другого стовпця таблиці, а висновок – у рядку 4 третього стовпця таблиці.

Подивимось, які додаткові твердження про значення істинності окремих змінних ми можемо зробити на основі відомих нам значень істинності складених формул.

Перш за все встановлюємо істинність формули $\neg X$, яка випливає з хибності X, і записуємо нове твердження у п'ятий рядок другого стовпця таблиці.

Продовжуючи міркування, робимо такий висновок. Якщо імплікація $\neg X' \to Y'$ (рядок 1) є тотожно істинною, і істинним є значення $\neg X'$ (рядок 5 таблиці), то істинним обов'язково має бути і значення Y' (згідно з таблицею 2 істинності імплікації, табл. 2), запис про що заносимо у шостий рядок другого стовпця нашої таблиці.

Відповідно хибним буде значення $\neg Y'$ (рядок 7 таблиці).

Але тотожна істинність формули $\neg Z' \vee \neg Y'$ (другий рядок таблиці) при хибності значення $\neg Y'$ (рядок 7), свідчить про істинність значення $\neg Z'$ (хоча б один член істинної диз'юнкції обов'язково має бути істинним згідно з її таблицею істинності). Заносимо це твердження у восьмий рядок другого стовпця нашої таблиці.

Далі, робимо висновок про хибність значення Z', що записуємо у дев'ятий рядок таблиці.

Але третій рядок таблиці містить твердження про істинність значення Z'!

Отже, ми отримали суперечність!

Тобто, одночасна істинність посилань і хибність висновку можуть бути отримані лише за умови одночасної істинності і хибності значення змінної Z, що забороняється законом суперечності логіки висловлень.

Отримана в процесі міркувань суперечність свідчить про те, що наше початкове припущення про неправильність аргументу (можливість одночасної істинності посилань і хибності висновку) було хибним, тобто даний аргумент є правильним.

Навпаки, якщо аргумент неправильний, то розмірковуючи як раніше, отримаємо такий набір значень змінних, для якого не виникне суперечності у випадку, коли формули, що відповідають посиланням, набудуть істинні значення (I), а формула, що відповідає висновку, набуде хибного значення (II).

Наприклад, перевіримо такий аргумент:

$$\begin{array}{|l} X \rightarrow (Y \vee Z) \\ Z \vee \neg Y \\ \hline (X \wedge Y) \vee Z \end{array}$$

Знов побудуємо таблицю, яку заповнимо на основі аналізу тверджень аргументу

Таблиця 5 – Результат скороченої перевірки неправильного аргументу

№ п/п	I	II	Коментар
1	$X' \rightarrow (Y' \vee Z')$		
2	$Z' \vee \neg Y'$		
3		$(X' \wedge Y') \vee Z'$	
4		$X' \wedge Y'$	з 3
5		Z'	з 3
6	$\neg Y'$		з 2 і 5
7		Y'	з 6
8		X'	з 1,5 і 7

Висновок про хибність кон'юнкції $(X' \wedge Y')$ і значення Z' випливає з припущення про хибність висновку $(X' \wedge Y') \vee Z'$ заданого аргументу, оскільки, згідно з таблицею істинності диз'юнкції (рис. 2), кожний член хибної диз'юнкції є хибним.

У свою чергу, істинність диз'юнкції $Z' \vee \neg Y'$ при хибності значення Z' свідчить про істинність значення $\neg Y'$ і, відповідно, хибність значення Y'.

Але імплікація $X' \rightarrow (Y' \vee Z')$ може бути істинною при хибності значень Y' і Z' лише за умови хибності значення X'.

Отриманий результат свідчить, що існує такий набір значень змінних $X'=0$, $Y'=0$, $Z'=0$, для якого посилання аргументу, що аналізується, будуть істинними, а висновок – хибним.

Дійсно, отримуємо:

$$
\begin{array}{lclcl}
X \rightarrow (Y \vee Z) & = & 0 \rightarrow (0 \vee 0) & = & 1\text{: істина} \\
Z \vee \neg Y & = & 0 \vee \neg 0 & = & 1\text{: істина} \\
(X \wedge Y) \vee Z & = & (0 \wedge 0) \vee 0 & = & 0\text{: хибність}
\end{array}
$$

Оскільки в даному прикладі вдалося знайти набір значень змінних X, Y і Z для якого всі посилання стають істинними, а висновок хибним, то розглянутий аргумент є неправильним.

4.4 Розв'язання логічних задач засобами логіки висловлень

Загальна схема розв'язання.

1. Вивчається умова задачі.
2. Вводиться система позначень для логічних висловлень.
3. Конструюється логічна формула, що описує логічні зв'язки між усіма висловленнями з умови задачі.
4. Визначаються значення істинності отриманої формули.
5. З отриманих значень істинності формули визначаються значення істинності введених логічних висловлень, на основі яких робиться висновок про розв'язок.

Задача 1. Троє уболівальників автогонок „Формула-1" сперечаються про результати майбутнього етапу:

„Шумахер не прийде першим", — каже Іван. „Першим буде Хіл".

„Ні, переможцем, як завжди, буде Шумахер, — заперечує Петро. — А про Алезі й казати нема чого, йому не бути першим".

Богдан, до якого звернувся Петро, не погоджується:

„Хілу не бачити першого місця, а ось Алезі пілотує найпотужнішу машину".

Після завершення етапу гонок виявилося, що кожне з двох припущень двох друзів підтвердилося, а обидва припущення третього друга виявилися хибними. Хто ж став переможцем?

Розв'язування. Формалізуємо логічні висловлення:

Ш — переможе Шумахер; Х — переможе Хіл; **А** — переможе Алезі.

Висловлення Богдана „Алезі пілотує найпотужнішу машину" не містить ніякого твердження про місце, яке посяде цей гонщик, тому в подальших міркуваннях не використовується.

Зафіксуємо висловлення кожного з уболівальників:

Іван: **¬Ш X**, Петро: **Ш ¬A**, Богдан: **¬X**.

Враховуючі, що припущення двох уболівальників підтвердилися, а третього – ні, запишемо і спростимо істинне висловлення:

$$(\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg (\neg X) \vee (\neg Ш \wedge X) \wedge \neg (Ш \wedge \neg A) \wedge \neg X) \vee$$
$$\vee \neg(\neg Ш \wedge X) \wedge (Ш \wedge \neg A) \wedge \neg X)$$

Неважко побачити, що перша і друга кон'юнкції будуть хибними оскільки в них зустрічаються співмножники $\neg Ш \wedge ... \wedge Ш$ і $\neg X \wedge ... \wedge X$, відповідно. Отже, отримаємо з третьої кон'юнкції:

$$(Ш \vee \neg X) \wedge Ш \wedge \neg A \wedge \neg X = (Ш \wedge \neg A \wedge \neg X) \vee (Ш \wedge \neg A \wedge \neg X) = Ш \wedge \neg A \wedge \neg X$$

Отримане висловлення істинне тільки за умов **Ш=1, А=0, Х=0. Відповідь.** Переможцем етапу став Шумахер.

Задача 2. Відомо, що у приладі **X** найчастіше виходять з ладу три вузли: **A, B, C**. З'ясувати, який саме вузол вийшов з ладу можна за трьома сигнальними лампочками **X, Y і Z** на контрольній панелі.

В інструкції з виявлення несправних вузлів записано:

1) якщо несправний хоча б один з вузлів приладу, то горить щонайменше одна з лампочок **X, Y, Z**;
2) якщо несправний вузол **A**, але справний вузол **C**, то світиться лампочка **Y**;
3) якщо несправний вузол **C**, але справний вузол **B**, світиться лампочка **Y**, але не світиться лампочка **X**;
4) якщо несправний вузол **B**, але справний вузол **C**, то світяться лампочки **X** і **Y** або не світиться лампочка **X**;
5) якщо світиться лампочка X і при цьому або несправний вузол **A**, або всі три вузли **A, B, C** працюють, то світиться і лампочка **Y**.

Припустимо, що на контрольній панелі засвітилася лампочка X. Згідно з інструкцією прилад було відремонтовано. Але, разом з тим, виявилося, що інструкція недосконала і можливі випадки, в яких вона не допоможе. Який вузол було замінено і які недоліки виявлені в інструкції?

Розв'язування. Нормалізуємо висловлення:

A — несправний вузол **A**; **X** — світиться лампочка **X**;
B — несправний вузол **B**; **Y** — світиться лампочка **Y**;
C — несправний вузол **C**; **Z** — світиться лампочка **Z**.

Тоді правилам 1 – 5 будуть відповідати такі формули:

(1) $A \vee B \vee C \to X \vee Y \vee Z$;
(2) $A \wedge \neg C \to Y$;
(3) $C \wedge \neg B \to Y \wedge \neg X$;
(4) $B \wedge \neg C \to (X \wedge Y \vee \neg X)$;
(5) $(A \vee \neg A \wedge \neg B \wedge \neg C) \wedge X \to Y$.

Оскільки формули 1 – 5 є істинними за умовою, істинною буде і їх кон'юнкція:

$$(A \vee B \vee C \to X \vee Y \vee Z) \wedge (A \wedge \neg C \to Y) \wedge (C \wedge \neg B \to Y \wedge \neg X) \wedge (B \wedge \neg C \to (X \wedge Y \vee \neg X)) \wedge$$
$$\wedge ((A \vee \neg A \wedge \neg B \wedge \neg C) \wedge X \to Y) = 1$$

Виразивши імплікацію через диз'юнкцію та заперечення отримаємо:

$$(\neg A \wedge \neg B \wedge \neg C \vee X \vee Y \vee Z) \wedge (\neg A \vee C \vee Y) \wedge (\neg C \vee B \vee Y \wedge \neg X) \wedge (\neg B \vee C \vee X \wedge Y \vee \neg X)$$
$$\wedge \wedge (\neg ((A \vee \neg A) \wedge (A \vee \neg B) \wedge (A \vee \neg C) \wedge X) \vee Y) =$$
$$(\neg A \wedge \neg B \wedge \neg C \vee X \vee Y \vee Z) \wedge (\neg A \vee C \vee Y) \wedge (\neg C \vee B \vee Y \wedge \neg X) \wedge (\neg B \vee C \vee X \wedge Y \vee \neg X)$$
$$\wedge \wedge (\neg A \wedge B \vee \neg A \wedge C \vee \neg X \vee Y) = 1.$$

Підстановкою до цієї тотожності конкретних значень істинності **X**=1, **Y**=0, **Z**=0, отримуємо:

$$(\neg A \vee C) \wedge (\neg C \vee B) \wedge (\neg B \vee C) \wedge (\neg A \wedge B \vee \neg A \wedge C) =$$
$$= (\neg A \wedge \neg C \vee \neg A \wedge B \vee C \wedge B) \wedge (\neg B \wedge \neg A \wedge C \vee C \wedge \neg A \wedge B \vee \neg A \wedge C) = \neg A \wedge B \wedge C = 1.$$

Звідси випливає, що **A**=0, **B**=1, **C**=1.

Відповідь на перше запитання задачі: необхідно замінити блоки **B** і **C**; блок **A** не потребує заміни.

Відповідь на друге запитання пропонується отримати самостійно.

4.5 Контрольні запитання

1. Що таке закони логіки? Які закони логіки Ви знаєте?
2. Сформулюйте закон контрапозиції і наведіть власний приклад його використання.
3. Які формули є двоїстими? Наведіть приклади двоїстих формул.
4. Чому в теоремі A→B, A є достатньою (але не є необхідною) умовою B, а B є необхідною (але недостатньою) умовою A.
5. Наведіть означення відношення логічного висновку.
6. Що гарантується під час використання правильного аргументу?

7. Нехай є два висловлення: А = „Цей ранок ясний" і В = „Цей ранок теплий". Опишіть такі формули звичайною мовою:

   а) $A \wedge B$; б) $A \wedge \neg B$; в) $\neg A \wedge \neg B$; г) $\neg A \vee B$; д) $A \vee \neg B$; е) $\neg A \vee \neg B$;

   ж) $\neg(A \wedge B)$; з) $\neg(A \vee B)$; и) $\neg(\neg A \wedge B)$; к) $A \to \neg B$; л) $\neg A \to B$;

   м) $\neg(A \to B)$.
8. З двох даних висловлень А і В побудуйте складене висловлення: а) істинне тоді і тільки тоді, коли обидва висловлення хибні;

   б) хибне тоді і тільки тоді, коли обидва висловлення істинні.
9. Визначте за допомогою таблиць істинності, які з нижченаведених формул є тотожно істинними або тотожно хибними:

   А) $\neg(\neg A \wedge B) \vee B((A \wedge B) \vee B)$; б) $((A \vee \neg B) \to B) \wedge (\neg A \vee B)$;

   в) $\neg(A \wedge B) \leftrightarrow (\neg A \vee \neg B)$; г) $A \wedge B \wedge (C \vee \neg E \vee D) \wedge \neg B$;

   д) $A \wedge (B \wedge \neg A \vee \neg B))$; е) $\neg((\neg A \vee B) \wedge \neg B \vee C)) \vee \neg A \vee C$;

   ж) $\neg((A \to B) \leftrightarrow (\neg B \to \neg A))$.
10. Спростіть нижченаведені формули з використанням законів склеювання

    а) $A \wedge B \wedge C \vee \neg A \wedge B \wedge C$; б) $A \wedge B \wedge C \vee A \wedge \neg(B \wedge C)$;

    в) $(A \vee B \vee C) \wedge (A \vee \neg B \vee C)$; г) $(\neg A \vee \neg B \vee \neg C) \wedge (\neg A \vee B \vee C)$.
11. Спростіть логічні формули:

    а) $A \wedge \neg C \vee C \wedge (B \vee \neg C) \vee (A \vee \neg B) \wedge C$;

    б) $\neg(A \wedge (B \vee \neg C) \vee \neg A \wedge B$;

    в) $(\neg A \vee C) \wedge (\neg A \wedge C) \wedge (B \vee \neg C) \wedge \neg(B \vee C)$;

    г) $A \wedge B \wedge C \vee A \wedge \neg B \wedge C \vee A \wedge B \wedge \neg C \wedge D$;

    д) $A \vee B \vee \neg B \wedge C \wedge D \vee \neg B \wedge \neg C \wedge \neg D \vee \neg B \wedge \neg C \wedge D$;

    е) $A \vee D \vee \neg A \wedge B \wedge C \vee \neg A \wedge \neg B \wedge C \vee \neg A \wedge \neg B \wedge \neg C$;

    ж) $\neg(A \vee B \vee C) \vee \neg B \vee (\neg(A \vee \neg B \vee C) \wedge \neg(\neg A \vee B \vee C) \vee \neg A \wedge \neg B$;

    з) $A \wedge B \wedge C \vee A \wedge B \wedge \neg C \vee A \wedge \neg B \wedge C \wedge D \vee A \wedge B \wedge C \wedge \neg D \vee A \wedge B \wedge C \wedge D$;

    и) $A \wedge D \wedge (\neg A \vee \neg C \wedge B \vee D) \vee A \wedge \neg C \vee \neg A \wedge B \wedge \neg C$.
12. Перевірте скороченим способом такий аргумент:

    „Для того, щоб скласти іспит, мені треба дістати підручник або конспект. Я дістану підручник тільки в тому випадку, якщо мій друг не виїде з міста. Він виїде тільки якщо я дістану конспект. Отже, я складу іспит".
13. Перевірте аргумент скороченим способом:

    „Для того, щоб бути допущеним до іспиту треба отримати залік. Я отримаю цей залік, якщо навчусь перевіряти аргументи скороченим способом. Я не засвоїв цей спосіб. Отже, мене не допустять до іспиту".
14. Троє дівчат – Роза, Маргарита і Лілія подали на конкурс вирощені ними рози, маргаритки і лілії. У жодної з дівчат ім'я не збігається з назвою улюблених квітів. Які квіти виростила кожна з дівчат?

15. Машина, що скоїла дорожно-транспортну пригоду зникла з місця події. Перший зі свідків сказав, що це були „Жигулі”, а перша цифра номера машини – одиниця. Другий свідок сказав, що це був „Москвич”, а номер починався з сімки. Третій – заявив, що то була іномарка, номер починався не з одиниці. При подальшому розслідуванні з'ясувалося, що кожний зі свідків правильно вказав або тільки марку машини, або тільки першу цифру номера. Якої марки була машина і з якої цифри починався номер?

5 ЧИСЛЕННЯ ПРЕДИКАТІВ ПЕРШОГО ПОРЯДКУ

5.1 Недостатність логіки висловлень

Засоби логіки висловлень дозволяють описувати та аналізувати далеко не будь-які міркування. Наприклад, за допомогою мови логіки висловлень неможливо виразити той факт, що з речення

«Є хоча б один студент, який розв'язав всі задачі»

випливає речення

«Кожну задачу розв'язав хоча б один студент».

Дійсно, ми можемо позначити перше твердження літерою A, а друге – літерою B. Але ми нічого не зможемо сказати про істинність або хибність імплікації $A \rightarrow B$, оскільки між антецедентом і консеквентом імплікації в даному випадку відсутній будь-який формальний зв'язок.

Можна формалізувати наведені твердження і інакше:

A = «Є хоча б один студент»
B = «Студент розв'язав всі задачі»
C = «Кожна задача розв'язана студентом».

При цьому отримаємо імплікацію:

$$(A \land B) \rightarrow (A \land C).$$

про тотожну істинність якої ми також не можемо судити.

Засобами логіки висловлень неможливо довести також правильність того, що з посилань

«A паралельно B» і
«B паралельно C»

випливає

«A паралельно C».

Дійсно, після формалізації в логіці висловлень цей аргумент набуде такого вигляду

$$\begin{array}{|l} X \\ \underline{Y} \\ Z \end{array}$$

Але формула

$$(X \wedge Y) \rightarrow Z$$

не є тавтологією.

Справа полягає в тому, що правильність цих аргументів визначається зв'язками, які існують між елементами внутрішньої структури елементарних висловлень. Але логіка висловлень не враховує цих зв'язків, оскільки елементарні висловлення є нероздільними і їх окремі частини в логіці висловлень не розглядаються. Це подібно до роздільної здатності мікроскопа, за межами якої ми вже не можемо роздивитися деталей (рис. 9).

Можливість урахування таких зв'язків дає мова логіки предикатів першого порядку, яка є розширенням логіки висловлень, тобто разом із усіма поняттями логіки висловлень містить також і ряд додаткових понять.

"Мала роздільна здатність" не дозволяє логіці висловлень зіставити три неідентичні між собою твердження

"Велика роздільна здатність" логіки предикатів, дозволяє зіставляти між собою окремі дев'ять елементів елементарних висловлень

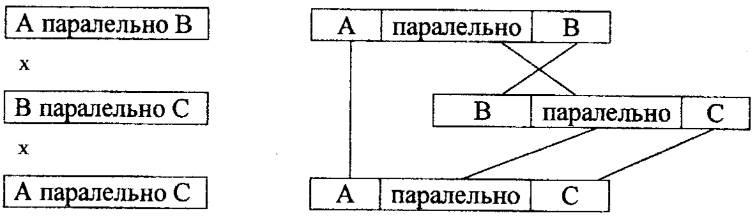


Рисунок 4 – Підвищення "роздільної здатності" логіки висловлень

Рисунок 4 ілюструє результат підвищення "роздільної здатності" логіки. При цьому кожне з трьох вихідних тверджень може бути подане вже не одним, а трьома окремими елементами. Умовно, кожне з тверджень в цьому випадку можна подати кон'юнкцією всіх трьох елементів:

$$A \wedge P \wedge B; \; B \wedge P \wedge C; \; A \wedge P \wedge C,$$

де символ P відображає властивість паралельності відповідних прямих.

При цьому відповідний аргумент набуде такого вигляду:

$$[(A \wedge P \wedge B) \wedge (B \wedge P \wedge C)] \rightarrow (A \wedge P \wedge C)$$

Неважко помітити, що наведена імплікація є тавтологією, оскільки хибність консеквентна обов'язково викличе й хибність антецедента. Дійсно, консеквент може стати хибним лише за умов хибності хоча б одного з членів кон'юнкції: A, P або C. Але хибність будь-якого з цих елементів призведе й до хибності кон'юнкції посилань (антецедента імплікації).

5.2 Поняття предиката. Предикати і висловлювальні форми

Розглянемо висловлювальну форму

$$\cos X = 1.$$

Кожному значенню змінної X на множині R дійсних чисел ця форма ставить у відповідність висловлення, тобто одне зі значень істинності (елемент множини {I,II}). Наприклад, будь-якому значенню X кратному 2 відповідає істинне висловлення cos X = 1, а всім іншим значенням X – хибне висловлення cos X = 1.

Таким чином, наведена висловлювальна форма задає відображення множини R дійсних чисел на множину {I,II}, тобто задає функцію з областю означення R і множиною значень {I,II}. Висловлювальні форми можуть задавати на R і такі функції, які завжди приймають значення I (наприклад, $\sin^2(X) + \cos^2(X) = 1$) або значення II (наприклад $|X| < 0$).

Предикатом називають логічну функцію, всі значення якої належать множині {I,II}. Предикат подається предикатним символом (предикатною буквою), за якою йде кортеж (впорядкована послідовність) з *n* термів. Предикат з *n* аргументами називають *n*-місним предикатом, наприклад одномісним, двомісним та ін.

Взагалі, у традиційній логіці в елементарному судженні відокремлюють суб'єкт та предикат. Суб'єкт є тим, про що йдеться у висловленні, а предикат – тим, що саме кажуть про суб'єкт. Наприклад, у висловленні

«Кішка має чотири лапи»

– кішка є суб'єктом, а «має чотири лапи» – предикатом. Якщо замінити «кішку» на «собаку», то знов отримаємо істинне висловлення. Але якщо за суб'єкт взяти «курка», то отримаємо хибне висловлення. Замінивши суб'єкт змінною, отримаємо висловлювальну форму «X має чотири лапи» і предикат як функцію, яку задає ця форма.

Одномісні предикати називають предикатами-властивостями, оскільки одномісна висловлювальна форма зі змінною, яка приймає значення з множини U, відображає властивість, притаманну елементам цієї множини.

Наприклад, предикат

високий (X),

відбере з множини U, лише ті елементи $X \in U$, для яких значення предиката високий (X) буде істинним, тобто лише ті елементи, які мають властивість "високий".

З іншого боку, n-місні предикати, $n>1$, називають предикатами-відношеннями, оскільки вони виділяють з деякої множини елементів саме ті елементи, які знаходяться між собою у відповідному відношенні.

Наприклад, предикат

брат (X, Y)

вибере з множини усіх людей лише такі їх пари (X, Y), які є братами, тобто знаходяться у родинних відношеннях братства.

Предикати, область означення яких є скінченною множиною, можуть задаватись за допомогою таблиць.

Кожному предикатові P, заданому на множині елементів M, відповідає множина IP його істинності, яка є підмножиною M і містить лише ті її елементи, яким відповідає істинне значення предиката P.

Якщо множина істинності предиката збігається з областю його означення, то предикат називають тотожно істинним.

Якщо множина істинності предиката є пустою, то предикат називають тотожно хибним.

Нехай є дві висловлювальні форми, однакові змінні в яких приймають значення з однієї множини. Такі висловлювальні форми називають *рівносильними*, якщо в будь-якому наборі значень змінних, що входять до їх складу, вони приймають однакові значення істинності.

Дві рівнозначні висловлювальні форми Ф1 і Ф2 з однаковими змінними, які розташовані в однаковому порядку, визначають один і той самий предикат, тобто, в цьому випадку для предикатів P1 і P2, що задані відповідно висловлювальними формами Ф1 і Ф2, виконується рівність P1= P2.

Висловлювальна форма Ф2 випливає з висловлювальної форми Ф1, якщо імплікація Ф1 $\rightarrow$ Ф2 перетворюється в істинне висловлення для будь-яких наборів значень змінних, що входять до її складу. Оскільки імплікація Ф1 $\rightarrow$ Ф2 перетворюється на хибне висловлення тільки у тих випадках, коли Ф1 перетворюється у істинне висловлення, а Ф2 – у хибне, то наведене означення можна переформулювати так: з Ф1 випливає Ф2, якщо завжди, коли Ф1 стає істинним висловленням, Ф2 також стає істинним висловленням.

Наприклад:

1) з рівності X = 0 випливає рівність X (X – 1) = 0;
2) з рівності (X – 1) X = 0 не випливає рівність X = 0, оскільки якщо X = 1 перша рівність є істинною, а друга – хибною;
3) з форми «X син A і B» випливає форма «A і B – батьки X», але з форми «A і B – батьки X» не випливає форма «V – син A і B», оскільки існує набір значень змінних (донька, батько, мати), для якого форма «A і B – батьки X» стає істинним висловленням, а форма «X – син A і B» – хибним;

4) з форми «V – парне» не випливає форма «X кратне 3», якщо M = N, і випливає, якщо M = {1;3;5;6;7;9;11;12}. Дійсно, у другому випадку кожен раз, коли істинною буде перша форма, істинною стане і друга форма.

З останнього прикладу можна побачити, що відношення логічного висновку між висловлювальними формами, як і відношення рівносильності, залежить, в загальному випадку, від множини, на якій вони розглядаються. На відміну від цих відношень у логіці висловлень, де змінні позначають довільні елементарні висловлення, тут мова йде про речення зі змінними, які замінюють деякий член речення (частіше за все підмет). При цьому з кожною змінною зв'язується множина її значень – чисел або інших об'єктів. Змінні такого типу називають предметними, тоді як змінні для висловлень називають висловлювальними або істиннісними змінними.

Відношення висновку між висловлювальними формами з предметними змінними пов'язано з відношенням включення між множинами істинності предикатів, які визначаються цими формами.

Якщо висловлювальні форми $\Phi 1(X_1, X_2, ..., X_n)$ і $\Phi 2(X_1, X_2, ..., X_n)$ визначають, відповідно, предикати P1 і P2, то умова

$$\Phi 1(X_1, X_2, ..., X_n) \to \Phi 2(X_1, X_2, ..., X_n)$$

виконується в тому і тільки в тому випадку, коли $P1 \to P2$.

Предикати зі своїми аргументами називаються атомами або атомарними формулами числення предикатів першого порядку. В загальному випадку атомом є розповідне речення, яке може бути або істинним, або хибним. Атом розглядається як єдине ціле, його структура і склад не аналізуються і не розкладаються на більш дрібні складові. Тобто, атомарна формула є елементарним об'єктом, який має істиннісне значення.

Для побудови атомів використовують предикатні та функціональні символи, змінні та константи. При цьому предикатні символи використовують для побудови висловлень (висловлювальних форм), в той час як функціональні символи використовують для побудови термів.

Наприклад, висловлення

«Син Романа є братом Івана»

можна подати за допомогою предиката таким чином:

брат (син (Роман, X), Іван),

де «брат» – предикатний символ, «син» – функціональний символ, «Роман», «Іван» –константи, «син (Роман, X)» – терм.

Ще раз зауважимо, що за «зовнішнім» виглядом цього терму не можна визначити, що він відображає: функцію чи предикат.

Для відображення фрази

«Мати Ольги одружена з батьком Ольги»

можна побудувати, наприклад, таку атомарну формулу:

подружжя (батько (Ольга), мати (Ольга)).

Предикати надають великі виразні можливості щодо відображення речень природної мови. Наприклад, фразу

«будинок жовтого кольору»

можна подати предикатом з одним, двома, трьома або більшою кількістю термів:

жовтий(будинок),
колір (будинок_1, жовтий),
значення(колір, будинок_1, жовтий).

Предикатний або функціональний символ часто називають функтором. Оскільки аргументи атомарної формули повинні бути розташовані в строго визначеному порядку, такі формули називають структурованими або просто структурами. Таким чином, структура складається з функтора, який подає ім'я відношення, і послідовності компонентів, які є об'єктами у відношенні. Кількість компонентів називають розмірністю або арністю структури.

5.3 Квантори загальності та існування

Як і в логіці висловлень з атомарних формул за допомогою логічних зв'язків можна будувати більш складні формули, які дозволяють формалізувати достатньо складні речення природної мови. Але використання тільки логічних зв'язків не дає зручних засобів для подання багатьох тверджень, наприклад таких, як «всі зебри смугасті».

Ми вже казали раніше, що перетворити висловлювальну форму у висловлення можна двома способами: підставити конкретні значення в змінні або використати спеціальні слова «будь-який» («кожний») і «існує» («деякі», «хоча б один»). Використання другого способу здійснюється за допомогою спеціальних конструкцій, які називаються кванторами.

Вираз «для будь-якого X» називається квантором загальності за змінною X (ясно, що замість X може бути використана будь-яка інша змінна). Цей вираз скорочено записується так:

$$\forall X(P(X)),$$

що означає – «для будь-якого [значення]X$P(X)$ – [істинне висловлення]». Слова в квадратних дужках часто опускаються.

Вираз «існує X, такий що...» називається квантором існування за змінною x і позначається $\exists X$. Запис $\exists X(P(X))$ означає, що «існує [значення] X таке, що $P(X)$ – [істинне висловлення]».

Потрібно звернути увагу на специфіку використання слова «деякий» в логіці та математиці. В звичайній мові, кажучи «деякий» мають на увазі «хоча б один, але не всі»; в логіці ж слово «деякі» означає «хоча б один, але, можливо, й усі».

Застосування до предиката $P(X)$ одного з кванторів називається операцією квантифікації. Внаслідок виконання цієї операції змінна в предикаті перестає бути змінною в звичайному розумінні цього слова, тобто символом, на місце якого можна підставити об'єкти з деякої множини. Змінна, до якої було застосовано операцію квантифікації, називається зв'язаною змінною, на відміну від звичайних змінних, які називають вільними змінними.

Якщо множина M значень змінної є скінченною, наприклад

$$M=\{a_1, A_2, ..., A_n\},$$

то:

- твердження $\forall X(P(X))$ має той же зміст, що і твердження

$$P(A_1)\wedge P(A_2)\wedge ...\wedge P(A_n);$$

- твердження $\exists X(P(X))$ має той же зміст, що і твердження

$$P(A_1)\vee P(A_2)\vee ...\vee P(A_n).$$

Множину значень квантифікованої змінної можна включати в запис квантора. Наприклад, твердження

«Для будь-якого натурального числа X виконується $1/X \leq X$» можна записати так:

$$(\forall X \subset N)(1/X \leq X).$$

Відмітимо також правомірність таких замін:

$$(\forall X \in M)\ (P(X)) \equiv \forall X(X \in M \rightarrow P(X));$$
$$(\exists X \in M)\ (P(X)) \equiv \exists X(X \in M \wedge P(X)).$$

Дійсно, перша рівносильність стверджує, що якщо будь-який елемент множини M має властивість P, то зі встановлення факту того, що елемент належить множині M випливає й те, що він має властивість P.

Інша справа, якщо у множині M існує елемент зі властивістю P. Тут вже не йдеться про наявність причинно-наслідкового зв'язку між належністю елемента множині M і наявністю в нього властивості P. В даному випадку можна лише стверджувати, що існує такий елемент, який водночас належить множині M і має властивість P.

Ще раз зверніть увагу на відмінність цих двох рівносильностей, і подумайте, чому в першій з них використовується операція імплікації, а у другій – операція кон'юнкції.

Внаслідок квантифікації по одній зі змінних n-місного предиката, відповідна n-місна висловлювальна форма стає (n–1)-місною висловлювальною формою. Оскільки одна зі змінних стає при цьому зв'язаною, то на її місце вже не треба підставляти ніяких значень.

Наприклад, вираз

$$X \times Y = 0$$

є двомісною висловлювальною формою з двома змінними: X і Y. Для судження щодо його істинності або хибності необхідно підставити значення на місця обох цих змінних.

В той же час, вираз

$$\forall X\ (X \times Y = 0)$$

є одномісною висловлювальною формою з однією вільною змінною Y і зв'язаною змінною X. Якщо $Y = 0$, ця форма стає істинним висловленням, якщо $Y \neq 0$, маємо хибне висловлення.

Можна зв'язати будь-яким з кванторів і змінну Y. В цьому випадку відразу отримаємо вираз з двома зв'язаними змінними і без жодної вільної змінної. Отже, не підставляючи ніяких значень на місця змінних, відразу отримаємо:

$$\exists Y \forall X\ (X \times Y = 0)$$

істинне висловлення (таким значенням Y є $Y = 0$), або

$$\forall X \forall X\ (X \times Y = 0)$$

хибне висловлення.

У загальному випадку для двомісного предиката можливе використання восьми комбінацій кванторів загальності та існування. На прикладі відношення

любить (X, Y),

де X – множина всіх чоловіків, а Y – множина всіх жінок, розглянемо більш детально графічну (рис. 5 – 8) і словесну інтерпретацію формул, в яких використовуються всі комбінації двох кванторів.

1. $\forall X \forall Y$ (P(X,Y)) – для будь-якого X і для будь-якого Y має місце (є істинним) відношення P (X,Y).

Словесна інтерпретація: «Усі чоловіки люблять усіх жінок» («Кожен чоловік любить кожну жінку»).

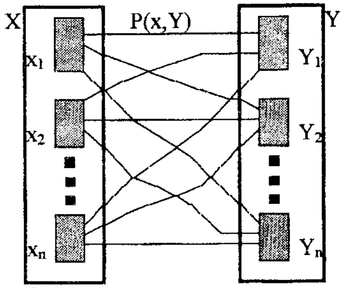


Рисунок 5 – Графічна інтерпретація формули ∀X∀Y (P(X,Y))

2. ∀Y∀X (P(X,Y)) – для будь-якого Y і для будь-якого X має місце P(X,Y).

Графічна інтерпретація цієї логічної формули збігається з інтерпретацією рис. 5 (за винятком напрямку стрілок, які на рисунку не показані).

Словесна інтерпретація: «Усі жінки люблять усіх чоловіків» («Кожна жінка любить кожного чоловіка»).

Зауважимо, що якщо б значення X і Y належали одному й тому ж самому домену (припустимо, домену люди), графічна і словесна інтерпретація першої і другої логічних формул повністю збігалися б.

3. ∃X∃Y (P(X,Y)) – існує X і існує Y такі, що має місце P(X,Y).

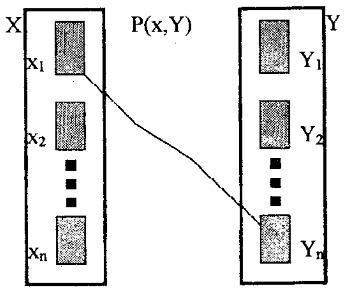


Рисунок 6 – Графічна інтерпретація формули ∃X∃Y (P(X,Y))

Словесна інтерпретація: «Існує [хоча б один] чоловік, який любить [хоча б одну] жінку».

4. ∃Y∃X (P(X,Y)) – існує Y і існує X, для яких має місце P(X,Y).

Графічна інтерпретація даної формули збігається з інтерпретацією рис. 6 (за винятком напрямку стрілки, який на рисунку не показаний).

Словесна інтерпретація: «Існує [хоча б одна] жінка яка любить [хоча б одного] чоловіка».

Знову бачимо, що якщо б значення X і Y належали одному й тому ж самому домену, графічна і словесна інтерпретація третьої і четвертої логічних формул повністю збігалися б.

5. $\forall Y \exists X\ (P(X,Y))$ – для будь-якого Y існує такий X, що має місце P(X,Y);

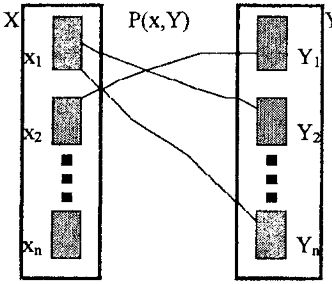


Рисунок 7 – Графічна інтерпретація формули $\forall Y \exists X\ (P(X,Y))$

Словесна інтерпретація: «Кожна жінка любить [хоча б одного] чоловіка».

Помітимо, що для різних Y може бути один і той самий X, але важливим є те, що кожному Y при цьому відповідає хоча б один X, який задовольняє відношення P(X,Y).

6. $\exists X \forall Y (P(X,Y))$ – існує X такий, що для будь-якого Y має місце P(X,Y);

Словесна інтерпретація: «Існує [хоча б один] чоловік, який любить усіх жінок».

Звернемо увагу на те, що зі сторони Y рисунки 7 і 8 є подібними, тобто, кожному *y* відповідає хоча б один X. Зі сторони ж X це не виконується: у першому випадку різні Y відповідають різним X, а у другому – усі Y відповідають одному X.

Отже, якщо істинним буде висловлення 6, обов'язково істинним буде і висловлення 5. Тобто, висловлення 5 випливає з висловлення 6. Дійсно, з того, що існує X, який задовольняє усі Y, випливає, що кожен Y задовольняється хоча б одним X.

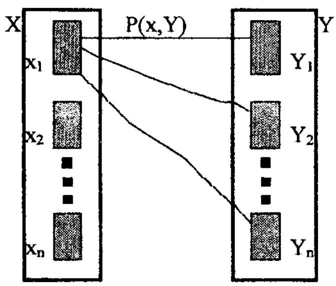


Рисунок 8 – Графічна інтерпретація формули $\exists X \forall Y\ (P(X,Y))$

7. $\forall X \exists Y\ (P(X,Y))$ – для будь-якого X існує Y такий, що має місце P(X,Y);

Графічна інтерпретація висловлення 7 аналогічна графічній інтерпретації висловлення 5 за умов заміни змінних X на Y і Y на X.

Словесна інтерпретація: «Кожен чоловік любить [хоча б одну] жінку».

8. $\exists Y \forall X(P(X,Y))$ – існує Y такий, що для будь-якого X має місце $P(X,Y)$;

Графічна інтерпретація висловлення 8 аналогічна графічній інтерпретації висловлення 7 за умов заміни змінних X наY і Y на X.

Словесна інтерпретація: «Існує [хоча б одна] жінка, яка любить усіх чоловіків».

Як і у випадку з висловленнями 5 і 6 бачимо, що якщо істинним буде висловлення 8, обов'язково істинним буде і висловлення 7. Тобто, висловлення 8 випливає з висловлення 7. Дійсно, з того, що існує Y, який задовольняє усі X, випливає, що кожен X задовольняється хоча б одним Y.

Підсумуємо наші спостереження. Відмітимо, що висловлення 1 і 2, та 3 і 4 мають один і той самий зміст, тобто одне й теж значення істинності.

Крім того, якщо істинним є висловлення 6, то істинним буде й висловлення 5, але не навпаки. Те ж саме стосується і висловлень 8 і 7.

Розглянемо ще один приклад. Якщо висловлення

$$\forall X \exists Y\ (X + Y = 0)$$

істинне, оскільки будь-яке число має протилежне число, то висловлення

$$\exists Y \forall X\ (X + Y = 0)$$

буде хибним, оскільки відсутнє таке число, яке було б протилежним до будь-якого іншого числа. Тобто, ще раз можемо впевнитися, що друге висловлення не випливає з першого (тобто висловлення типів 6 і 8 не випливають з висловлень типів 5 і 7, відповідно).

Зробимо висновки:

- однойменні квантори можна міняти місцями;
- різнойменні квантори не можна міняти місцями;
- якщо предикат $\exists X \forall Y\ (P(X,Y))$ – істинний, то істинним буде й предикат $\forall Y \exists X\ (P(X,Y))$. Для останньої залежності не важлива множина, на якій розглядаються речення, вона визначається лише їх логічною формою.

Висловлювальні форми (предикати), в яких всі змінні є квантифікованими, перетворюються на висловлення, які також називають реченнями. При цьому характерним признаком речень (висловлень) є те, що їм можна однозначно ставити у відповідність певне значення: істина або хибність, – в той час як для предикатів, які містять неквантифіковані змінні, це не виконується. Наприклад, предикатна формула БАТЬКО (X,Y) не є висловленням. Для одних X її значенням буде істина, а для інших– хибність, в залежності від того, які значення будуть підставлені на місця змінних. В протилежність цьому, значення істинності предикатної формули із зв'язаними змінними можна визначити, не виконуючи таких підстановок. Наприклад, висловлення

$(\forall Y)(\exists X)$ БАТЬКО(X,Y) («Кожна людина Y має батька X»)

є істинним, в той час як висловлення

$(\forall X)(\exists Y)$ БАТЬКО(X,Y) («Усі люди мають дітей»)

є хибним.

5.4 Заперечення речень з кванторами

Відомо, що для заперечення деякого речення достатньо розмістити перед його присудком частку «не». Наприклад, речення

«Я люблю каву»

«Я не люблю каву»

заперечують одне одного, оскільки мають різні значення істинності.

Подивимось, як саме будуються заперечення речень з кванторами. Розглянемо такі два речення:

«Всі птахи літають»

«Всі птахи не літають»

На відміну від попередніх, ці речення не є запереченнями одне одного, тому що обидва вони помилкові.

Також не є запереченнями одне одного і речення:

«Деякі птахи літають»

«Деякі птахи не літають»,

оскільки обидва вони є істинними.

Таким чином, речення, отримані додаванням частки «не» до присудка речень вигляду «Всі X суть P» і «Деякі X суть P», не є запереченнями цих речень. У даному випадку заперечення будується за допомогою словосполучення «неправильно, що».

Дійсно, запереченням речення

«Всі птахи літають»

буде речення

«Неправильно, що всі птахи літають».

Але воно має той самий зміст, що і речення

«Деякі птахи не літають».

Отже, в даному випадку маємо рівнозначність:

$$\neg\{(\forall X)P(X)\} \leftrightarrow (\exists X)\{\neg P(X)\}.$$

Аналогічно, запереченням речення «Деякі птахи літають» буде речення «Неправильно, що деякі птахи літають», яке має той же зміст, що і речення «Всі птахи не літають».

Тобто, отримуємо

$$\neg\{(\exists X)\, P(X)\} \leftrightarrow (\forall X)\{\neg P(X)\}.$$

В зв'язку з цим квантори загальності та існування називають двоїстими один одному.

Для побудови заперечення речення, яке починається з декількох кванторів, наприклад,

$$(\forall X)(\exists Y)(\forall Z)P(X,Y,Z)$$

необхідно послідовно використати наведене правило до кожного з кванторів формули. При цьому отримаємо:

$$\neg\{(\forall X)(\exists Y)(\forall Z)\, P(X,Y,Z)\} \leftrightarrow (\exists X)\, \neg\{(\exists Y)(\forall Z)\, P(X,Y,Z)\} \leftrightarrow$$
$$\leftrightarrow (\exists X)\,(\forall Y)\, \neg\{(\forall Z)\, P(X,Y,Z)\} \leftrightarrow (\exists X)\,(\forall Y)(\exists Z)\,\{\neg P(X,Y,Z)\}.$$

Тобто, має місце таке загальне правило:

для того, щоб побудувати заперечення речення, яке розпочинається з квантора, достатньо кожен квантор речення замінити на його двоїстий квантор, а заперечення перенести на речення, яке розміщується за кванторами.

5.5 Контрольні запитання

1. Наведіть означення поняття „предикат”.
2. Які переваги має логіка предикатів у порівнянні з логікою висловлень?
3. Чим принципово відрізняються змінні у логіці предикатів від змінних у логіці висловлень?
4. Які логічні зв'язки дозволяють відобразити дію кванторів загальності та існування?
5. Які квантори можна міняти місцями у логічній формулі, а які не можна?
6. Визначте, які зі змінних у наведеній формулі є вільними, а які зв'язаними:

   $$Y\, P(X, Y)\ \&\ \neg.X\, P(X,X)?$$

7. Зобразіть предикатною формулою речення:

   „Усі прості числа перевищують X”.

8. Чи будуть рівносильними такі пари формул:

   а) $(X)(F(X)\vee G(X))$ і $(X)F(X)\vee(X)G(X)$;

   б) $(X)(F(X)\&G(X))$ і $(X)F(X)\&(X)G(X)$;

   в) $(X)(F(X)\rightarrow G(X))$ і $(X)F(X)\rightarrow(X)G(X)$;

г) $(X)F(X)\to(X)G(X)$ і $(X)(Y)(F(X)\to G(Y))$;

д) $(X)(F(X)\to G(X))$ і $(X)F(X)\to(X)G(X)$;

е) $(X)F(X)\to(X)G(X)$ і $(X)(Y)(F(X)\to G(Y))$;

ж) $(X)(F(X)\leftrightarrow G(X))$ і $(X)F(X)\leftrightarrow(X)G(X)$;

з) $(X)(F(X)\leftrightarrow G(X))$ і $(X)F(X)\leftrightarrow(X)G(Y)$.

9. На множині M задано одномісний предикат P(X). За допомогою предикатів з кванторами відобразіть такі твердження:

   а) існує не менше одного елемента, що задовольняє P(X);

   б) існує не більше одного елемента, що задовольняє P(X);

   в) існує точно один елемент X, що задовольняє P(X).

10. Сформулюйте заперечення таких висловлень в стверджувальній формі (тобто, заперечення даного висловлення не повинно починатись зі слів «неправильно що» та «не»):

    а) «В кожному місті є район, в кожній школі якого є клас, в якому ні один учень не займається спортом»;

    б) «Існує книга, в якій є сторінка, в кожному рядку якої знайдеться хоча б одна буква «а».

6 ІНТЕРПРЕТАЦІЯ ФОРМУЛ ЧИСЛЕННЯ ПРЕДИКАТІВ ПЕРШОГО ПОРЯДКУ

6.1 Поняття правильно побудованої формули числення предикатів

Поняття формули логіки предикатів наведемо аналогічно тому, як це було зроблено в логіці висловлень.

Спочатку задамо алфавіт символів, з яких можна будувати формули:

- предметні змінні: X, Y, Z, X_i, Y_i, Z_i (i – натуральне число);
- предикатні букви: P, Q, R, P_i, Q_i, R_i (i – натуральне число);
- символи операцій: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$;
- символи кванторів $\forall$ і $\exists$;
- допоміжні символи: (,) – дужки і кома.

Вирази вигляду T і $T(X_1,X_2,...,X_n)$, де X_i – предметні змінні, – називають відповідно, нульмісними, та n-місними предикатними символами.

Тепер можна дати таке означення формули логіки предикатів.

1. Будь-який нульмісний предикатний символ є формулою.
2. Будь-який n-місний предикатний символ є формулою.
3. Якщо F – формула, а X – предметна змінна, то вирази $\forall X(P(X))$ і $\exists X(P(X))$ є формулами.
4. Якщо F1 і F2 – формули, то формулами будуть і $\neg F1$, $F1 \wedge F2$, $F1 \vee F2$, $F1 \rightarrow F2$, $F1 \leftrightarrow F2$.
5. Ніяких інших формул в логіці предикатів не існує.

Формули, які визначені в пп. 1 і 2, називаються атомарними формулами. Формули, які побудовані з атомарних формул шляхом квантифікації або використання логічних зв'язок відповідно до пп. 3 та 4 даного означення, називаються правильно побудованими формулами логіки предикатів, що скорочено позначаються як ппф.

Наприклад, формулами логіки предикатів є такі формули:

$$\forall X(P(X,Y)),\ Q(U),\ R(X) \wedge \forall XQ(X,Y,Z),\ \neg R(Z).$$

Нагадаємо, що якщо всі входження змінної в формулу знаходяться в області дії квантора за цією змінною, або вона входить в цей квантор, то така змінна називається зв'язаною у формулі.

Змінна називається вільною у формулі, якщо хоча б одне її входження в цю формулу є вільним.

Отже, змінна X у першій з наведених формул є зв'язаною, а у третій – вільною, оскільки існує її входження в формулу (R(X)), яке не підпадає під область дії жодного квантора.

6.2 Інтерпретація правильно побудованих формул

Значення істинності будь-якої правильно побудованої формули (ппф) як в логіці висловлень, так і в логіці предикатів, можна встановити тільки з урахуванням конкретної *інтерпретації.* Інтерпретація ппф у логіці висловлень полягає в присвоєнні істиннісних значень кожному атомарному висловленню, яке входить до ппф. Інтерпретація ппф в логіці предикатів є значно більш складною, оскільки тут атомарні формули містять терми. При цьому для інтерпретації формули необхідно визначити непусту множину M значень, які можна приписувати предметним змінним, що входять у формулу, а кожному функціональному та предикатному символам – поставити у відповідність функцію та відношення, визначені на множині M. Нульмісним предикатам незалежно від вибраної множини M приписується одне зі значень істинності: I або II.

Таким чином, інтерпретацією *I* числення предикатів з областю інтерпретації (носієм) M називається набір функцій, який зіставляє:

- кожну предметну константу *a* з елементом носія $I(a)$, $I(a) \in M$;
- кожен n-місний функціональний символ *f* з операцією $I(f)$ на носії,
$$I(f)\colon M^n \to M;$$
- кожну n-місну предиката P з відношенням $I(P)$ на носії,
$$I(P) \in M^n.$$

Розглянемо декілька прикладів інтерпретації.

Нехай задано множину S формул і множину D об'єктів деякої предметної області, яка інтерпретується за допомогою S.

Тоді, для здійснення інтерпретації, необхідно виконати такі зіставлення:

- кожній константі поставити у відповідність елемент з предметної області D;
- кожному n-місному функціональному символу поставити у відповідність відображення елемента з D^n у D;
- кожному n-місному предикатному символу поставити у відповідність відображення елемента з D^n у множину {I,II}.

Після здійснення таких операцій можна казати, що деяке речення S істинне або хибне в залежності від того, істинним чи хибним буде відповідне висловлення про D.

Розглянемо формулу

$$(\forall X)(\exists Y)A(X,Y),$$

яку задано на області D=(0,1), і предикат A(X,Y), який на цій області набуває значень, показаних на рисунку 9:

X	0 0 1 1
Y	0 1 0 1
A(X,Y)	I II I II

Рисунок 9 – Таблиця значень предиката A(X,Y)

Наведена формула буде істинною, оскільки для кожного X в області D {0,1} знайдеться такий Y (у нашому прикладі Y = 1), що буде виконуватися A (X, Y) ∈ I (дійсно, A (0, 1) = I і A (1, 1) = I).

Ще один приклад. Роз'яснемо формулу

$$\exists Y \forall X\ P(X,Y) \rightarrow (A(X) \wedge B),$$

яка містить вільну змінну X.

Для цього:

- задамо область M означення змінних X і Y: M={1,2};
- задамо відображення двомісного предикатного символу P (рис. 10, а) і одномісного предикатного символу A (рис. 10, б) на множину {I,II}:

X	1	1	2	2
Y	1	2	1	2
P(X,Y)	I	II	I	II

а)

X[	1	2
A(X)	I	II

б)

Рисунок 10 – Таблиці значень предикатів: а) P(X,Y); б) A(X)

- нульмісному предикатному символу (константі) B припишемо значення I;
- вільному входженню змінної x припишемо значення 1.

Для заданого інтерпретації подана формула перетворюється на істинне висловлення.

Але для інших інтерпретацій задана формула може приймати й помилкові значення. Це станеться, наприклад, для таких інтерпретацій, у яких змінній X буде приписано значення 2 або символу B – значення II, або предикатній букві A – предикат «парний».

Розглянемо ще один приклад.

Нехай предметна область – родинні відношення між деякою особою «Роман» і деякими іншими особами. Областю інтерпретації є множина людей: {Роман, Олег – двоюрідний брат Романа; Іван – двоюрідний брат Романа; Катерина – подружка Івана}.

Визначимо функції та відношення, які мають місце на даній предметній області. Тут можна визначити таку функцію з одним аргументом, як «подружка». Конкретне застосування даної функції у нашому прикладі відображає елемент «Іван» (аргумент) на елемент «Катерина» (значення функції). На даній предметній області можна також визначити відношення «двоюрідний брат», яке має два аргументи і відображає два імені людей на одне істиннісне значення. Так, пару імен Роман і Олег буде відображено на значення І, оскільки вони дійсно є двоюрідними братами. В той же час пару імен Роман і Катерина буде відображено на значення ІІ, оскільки вони не є двоюрідними братами.

Виберемо позначення для констант і припишемо їм значення, які відповідають елементам області інтерпретації:

A – двоюрідний брат Романа – Олег;

B – сам Роман;

C – двоюрідний брат Романа – Іван.

Одноаргументну функцію «подружка» позначимо символом *f*. Семантично ця функція визначається відображенням того, що значенням *f(C)* є Катерина.

Відношення «двоюрідний брат» позначимо символом *P* з двома аргументами. Тоді семантичне означення цього відношення задається переліком його значень, наведених на рисунку 11:

P(A,B)	P(B,A)	P(C,B)	P(B,C)	P(A,C)	P(C,A)	P(A,f(C))	P(f(C),A)	P(B,f(C))	P(f(C),B)	P(C,f(C))	P(f(C),C)
I	I	I	I	II	II	II	II	II	II	II	II

Рисунок 11 – Семантичне означення відношення P

Бачимо, що для інтерпретації кожен конкретний випадок відношення «двоюрідний брат» має бути зображений атомарною формулою з визначеним істиннісним значенням.

З наведених атомарних формул можна побудувати різноманітні правильно побудовані формули, наприклад:

$$P(A,B) \wedge \neg P(B,f(C))$$

«Олег двоюрідний брат Романа, а Роман не є двоюрідним бартом Катерини» I (істина),

$$\exists X P(X,B)$$

«існує деяка особа, яка є двоюрідним братом Романа» I (істина),

$$\forall X P(X,B)$$

«будь-яка людина є двоюрідним братом Романа» II (хибність).

Множина істинних ппф, створених описаним способом, що мають значення «істина», будуть теорією даної предметної області, а кожна окрема ппф – аксіомою. Якщо складена теорія буде адекватно описувати предметну область, то всі факти про неї, які є істинними, будуть випливати з аксіом цієї теорії, і жоден хибний факт не буде випливати з них. Теорія буде повною, якщо всі істинні факти предметної області будуть випливати з неї.

Зауважимо, що якщо для логіки висловлень завжди можна з'ясувати за допомогою таблиці істинності чи є ппф тавтологією, то для логіки предикатів такого загального способу означення загальнозначимості не існує. Тобто, загальнозначимість деяких формул можна встановити лише шляхом міркувань.

Розглянемо, наприклад, формулу

$$\forall X P(X) \rightarrow P(Y).$$

Припустимо, що на деякій множині M, $X,Y \in M$, букві P поставлено у відповідність такий предикат P(Y), який на деякому значенні Y набуває значення «хибність». Тобто, у множині M існує такий елемент, який не має властивості P. Але при цьому і вираз $\forall X P(X)$ отримає значення «хибність», оскільки не кожний елемент множини M має властивість P. Отже, розглянута формула є загальнозначимою.

Для доведення того, що формула не є загальнозначимою, достатньо вказати хоча б одну інтерпретацію, в якій вона прийме значення «хибність».

Розглянемо формулу

$$\exists X P(X) \rightarrow \forall X P(X).$$

Нехай $M=\{3;4\}$, а P – предикат «просте число». Тоді $\exists X P(X)$ – прийме значення «істина», а $\forall X P(X)$.– значення «хибність», тобто вся формула набуде значення «хибність».

6.3 Формалізація елементарних висловлень з урахуванням їх внутрішньої структури

Логіка предикатів забезпечує можливість формалізації елементарних висловлень з урахуванням їх внутрішньої структури. Наприклад, елементарне висловлення

«У кожної людини є мати»

можна формалізувати за допомогою предиката МАТИ (X,Y), який визначається на множині людей:

$$\forall Y \exists X \text{ МАТИ } (X, Y).$$

Якщо ввести додатково предикат «бути людиною», то наведене висловлення можна подати у вигляді:

$$\forall Y\ (\text{ЛЮДИНА}\ (Y) \rightarrow \exists X\ \text{МАТИ}\ (X, Y)).$$

Розглянемо два речення:

«В Києві живе чоловік, який має сестру у Вінниці»;

«У Вінниці живе жінка, яка має брата в Києві».

Неважко зрозуміти, що вони рівнозначні (кожне з них випливає з іншого), але на рівні логіки висловлень, їх формалізація не виявить цієї рівносильності. На цьому рівні кожне речення формалізується або як елементарне висловлення (A, B), або як кон'юнкція двох елементарних висловлень:

$$A1 \wedge A2,$$
$$B1 \wedge B2,$$

де A1 – «жінка живе у Вінниці»;
A2 – «жінка має брата в Києві»;
B1 – «чоловік живе в Києві»;
B2 – «чоловік має сестру у Вінниці».

Але, формули A і B, як і формули $A1 \wedge A2$ і $B1 \wedge B2$, не випливають одна з одної.

Формалізація ж на рівні логіки предикатів виявляє рівносильність даних речень. Введемо позначення:

A1(X) – «X – жінка»;
A2(X) – «X – живе у Вінниці»;
B1(X) – «X – чоловік»;
B2(X) – «X – живе в Києві»;
C(X,Y) – «X – мати Y».

Тоді висловленню

«У Вінниці живе жінка, яка має брата в Києві»

буде відповідати формула

$$\exists X(A1(X) \wedge A2(X) \wedge \exists Y(B1(Y) \wedge B2(Y) \wedge \exists Z(C(Z,X) \wedge C(Z,Y)))),$$

яка повністю рівносильна формулі

$$\exists Y(B1(Y) \wedge B2(Y) \wedge \exists X(A1(X) \wedge A2(X) \wedge \exists Z(C(Z,X) \wedge C(Z,Y)))),$$

що відповідає висловленню

«В Києві живе чоловік, який має сестру у Вінниці».

За допомогою логіки предикатів можна подати і досить складні речення. При цьому слід дотримуватись певних загальних правил. Наприклад, слід задавати окремий предикат для області означення кожної змінної, уважно ставитися до використання зв'язок імплікації та кон'юнкції, місця розташування кванторів загальності та існування.

Розглянемо декілька прикладів такої формалізації.

1. «Всі кубики, які знаходяться на кубиках, що були пересунуті або були з'єднані з кубиками, що пересувались, теж були пересунуті»

Спочатку визначаємо, що під час формалізації необхідно буде описати взаємне розташування двох кубиків. Отже, необхідно використати дві змінні та ввести предикат КУБИКИ для визначення предметної області, якій належать ці змінні. Для описання взаємного розташування кубиків необхідно ввести предикати ЗВЕРХУ і ЗЧЕПЛЕНІ, а для описання дій над кубиками, предикат ПЕРЕСУНУТИЙ. Оскільки твердження справедливе для будь-яких кубиків, обидві змінні будуть зв'язані кванторами загальності, областю дії яких буде вся формула. Причинно-наслідковий зв'язок відобразиться за допомогою імплікації. Отже, перше твердження може бути подане такою формулою:

$$\forall X \forall Y \{[\text{ Кубик } (X) \wedge \text{ Кубик } (Y) \wedge (\text{Зверху } (X, Y) \vee \text{ З'єднані } (X, Y)) \wedge\wedge \text{ Пересунутий } (Y)\ ] \rightarrow \text{ Пересунутий } (X)\ \}$$

Наведемо словесне описання отриманої формули, але більш формалізоване, ніж вихідне речення:

"ДЛЯ БУДЬ-ЯКИХ *X* і *Y* , ***<u>якщо</u>*** *X* Є КУБИКОМ ***і*** *Y* Є КУБИКОМ, ***і*** *X* знаходиться ЗВЕРХУ *Y*, ***або*** *X* З'ЄДНАНИЙ з *Y*, ***і*** *Y* ПЕРЕСУНУТИЙ, ***<u>то</u>*** і *X* ПЕРЕСУНУТИЙ"

У даному твердженні великими буквами позначені квантори і предикати, напівжирним курсивом – логічні зв'язки, а звичайним курсивом – змінні. Слова, що відносяться до імплікації додатково підкреслені.

2. «Для кожної множини X існує множина Y така, що потужність Y більша за потужність X»

У даному реченні присутні чотири об'єкти, два з яких є множинами, а два – потужностями. Отже, для означення доменів, яким належать ці змінні, необхідно ввести відповідні предикати МНОЖИНА та ПОТУЖНІСТЬ. Крім того, слід використати предикат БІЛЬШЕ. Очевидно, що кількість змінних складе чотири (за кількістю об'єктів).

Перефразуємо вихідне речення з урахуванням наведеного вище:

"ДЛЯ БУДЬ-ЯКОГО *X*, ***якщо*** *X* є МНОЖИНОЮ, ***то*** ІСНУЄ *Y* такий, що *Y* є МНОЖИНОЮ, ***і*** ІСНУЮТЬ *U* і *V* такі, що *U* є ПОТУЖНІСТЮ множини *X*, а *V* є ПОТУЖНІСТЮ множини *Y*, ***і*** *V* є БІЛЬШИМ за *U*"

Таке твердження може бути формалізовано в такому вигляді:

$$\forall X\{\text{Множина}(X) \rightarrow \exists Y[\text{Множина}(Y) \wedge \exists U\exists V[\text{Потужність}(X, U) \wedge \wedge\ \text{Потужність}(Y, V) \wedge \text{Більше}\ (U, V)]]\}$$

А чому б не формалізувати твердження у такому вигляді:

$$\forall X\exists U\{[\text{Множина}(X) \wedge \text{Потужність}(X, U)] \rightarrow \exists Y\exists V[\text{Множина}(Y) \wedge \wedge\ \text{Потужність}\ (Y, V) \wedge \text{Більше}(U, V)]]\}?$$

Але у даному випадку природною мовою речення буде подане в такому вигляді:

"***Якщо*** ДЛЯ БУДЬ-ЯКОГО *X* ІСНУЄ *U* такий, що *X* є МНОЖИНОЮ і *U* є ПОТУЖНІСТЮ множини *X*, ***то*** ІСНУЮТЬ *Y* і *V* такі, що *Y* є МНОЖИНОЮ ***і*** *V* є ПОТУЖНІСТЮ множини *Y*, ***і*** *V* є БІЛЬШИМ за *U*"

Очевидно, що перший варіант більше відповідає семантиці вихідного речення, підкреслюючи саме той факт, що для будь-якої множини завжди можна знайти іншу множину з більшою потужністю.

$$\forall X\{\text{Множина}(X)\rightarrow\exists Y\exists U\exists V\ [\text{Множина}(Y)\wedge\text{Потужність}(X,U)\wedge\text{Потужність}(Y,V)\wedge\text{Більше}(U,V)]\}$$

3. «В кожному місті є собаколов, якого покусав кожен собака міста» – "ДЛЯ БУДЬ-ЯКОГО *X*, ***якщо*** *X* є МІСТОМ ***то*** ІСНУЄ *Y* таке, що *Y* є СОБАКОЛОВОМ ***і*** *Y* ЖИВЕ В *X*, ***і*** ДЛЯ БУДЬ-ЯКОГО *Z*, ***якщо*** *Z* є СОБАКОЮ і *Z* ЖИВЕ в *Y*, ***то*** *Z* ПОКУСАВ *Y*"

$$\forall X\{\text{Місто}(X)\rightarrow\exists Y[\text{Собаколов}(Y)\wedge\text{Жити\_в}(X,Y)\wedge\forall Z\{[\text{Собака}(Z)\wedge\text{Жити\_в}(X,Z)]\rightarrow\text{Кусав}(Y,Z)\}]\}$$

6.4 Контрольні запитання

1. Наведіть означення правильно побудованої формули логіки предикатів.
2. Які дії необхідно виконати для інтерпретації ппф у логіці предикатів?
3. Які способи визначення істиннісного значення формули використовуються у логіці висловлень і у логіці предикатів?
4. Нехай $F(X)=P(X)\&(\forall Y)(P(Y)\rightarrow D(X,Y))$, $M=\{2,3,4,6,9\}$. Знайдіть предикати, які відповідають F(X) в таких інтерпретаціях:
   а) P(X) = «X – просте число», D(X,Y) = «X менше або дорівнює Y»;

б) P(X) = «X – непарне число», D(X,Y) = «X ділить Y»;

в) P(X) = «X не дорівнює 4», D(X,Y) = «X менше або дорівнює Y».

Чи існує інтерпретація, при якій формулі F(X) відповідають предикати?

а) «X = 4»;

б) «X – парне».

5. Нехай M – множина всіх точок, прямих і площин тривимірного простору. Розглянемо алгебраїчну систему <M; X∈Y, p(X), l(X), pl(X)>, де ∈ – відношення належності; p(X) означає, що X є точка; l(X) – X є пряма; pl(X) – X є площина. Зобразіть нижченаведені предикати з використанням указаної сигнатури:

   а) «площини X і Y мають спільну точку»;

   б) «якщо площини X і Y мають спільну точку, то вони мають спільну пряму»;

   в) «прямі X і Y мають спільну точку»;

   г) «прямі X і Y паралельні»;

   д) «прямі X, Y і Z утворюють трикутник».

6. Підберіть сигнатуру і подайте нижченаведені міркування у вигляді послідовності формул логіки предикатів:

   а) деякі з першокурсників знайомі зі всіма другокурсниками, а деякі з другокурсників – спортсмени. Отже, певна група першокурсників знайомі з деякими спортсменами;

   б) членом правління клубу може бути кожен повнолітній член клубу. Ігор і Андрій – члени клубу. Ігор повнолітній, а Андрій старший за Ігоря. Отже, Андрій може бути членом правління клубу;

   в) митники обшукують кожного, хто приїздить до країни, крім високопосадових осіб. Деякі люди, які сприяли провезенню до країни наркотиків, приїздили до країни й були обшукані виключно людьми, які також сприяли провезенню наркотиків. Ніхто з високопосадових осіб не сприяв провезенню наркотиків. Отже, деякі з митників сприяли провезенню наркотиків.

7 НОРМАЛІЗАЦІЯ ФОРМУЛ ЧИСЛЕННЯ ПРЕДИКАТІВ ПЕРШОГО ПОРЯДКУ

7.1 Перетворення правильно побудованих формул до клаузальної нормальної форми

Логіка предикатів, подібно до природної мови, дає можливість виразити вміст одного й того ж самого висловлення декількома різними способами, в залежності від досвіду та звичок конкретної людини. В зв'язку з цим, перед виконанням формальної процедури автоматичного логічного виведення, виникає необхідність перетворення ппф, які задані в довільній формі у деяку стандартну нормальну форму, зручну з точки зору функціонування механізму виведення.

В логіці предикатів першого порядку стандартною формою подання речення є диз'юнкція літералів. Літералом називається атомарна формула або її заперечення. Диз'юнкція літералів називається диз'юнктом або реченням. Диз'юнкція є істинною, якщо істинним є хоча б один літерал, що входить до її складу. Диз'юнкт, який містить n літералів називається n-літеральним диз'юнктом. Диз'юнкт називається пустим, якщо він не містить таких літералів, які можуть стати істинними в жодній інтерпретації.

Будь-яка ппф числення предикатів може бути перетворена на множину речень (тобто, диз'юнкцій літералів). Проілюструємо процес такого перетворення на прикладі ппф:

$$\forall X\{P(X)\to\{\forall Y\{P(Y)\to P(f(X,Y))\}\land\neg\forall Y\{Q(X,Y)\to P(Y)\}\}\}.$$

Процес перетворення складається з дев'яти основних етапів.

1. ***Вилучення символів імплікації.***

Всі входження символу імплікації заміщуються підстановкою

$$\neg X1\lor X2 \text{ замість } X1\to X2$$

по всій формулі. Отримуємо:

$$\forall X\{\neg P(X)\lor\{\forall Y\{\neg P(Y)\lor P(f(X,Y))\}\land\neg\forall Y\{\neg Q(X,Y)\lor P(Y)\}\}\}.$$

2. ***Обмеження області дії символу заперечення.***

Кожний знак заперечення повинен застосовуватися не більше як до однієї атомарної формули, що забезпечується завдяки використанню таких підстановок:

$$\neg(A\lor B)\leftrightarrow \neg A\land\neg B;$$
$$\neg(A\land B)\leftrightarrow \neg A\lor\neg B;$$
$$\neg\neg A\leftrightarrow A.$$

$\neg\{\forall X A(X)\} \leftrightarrow \exists X\{\neg A(X)\}$ – якщо неправильно, що А виконується для будь-якого Х, то існує такий Х, для якого А не виконується.

$\neg\{\exists X A(X)\} \leftrightarrow \forall X\{\neg A(X)\}$ – якщо не існує жодного Х такого, щоб виконувалося А, то для будь-якого Х – А не виконується.

Для нашого прикладу отримаємо:

$$\forall X\{\neg P(X\vee\{\forall Y\{\neg P(Y)\vee P(f(X,Y))\}\wedge\exists Y\{Q(X,Y)\wedge\neg P(Y)\}\}\}$$

3. ***Розподіл (стандартизація) змінних.***

В області дії будь-якого квантора, пов'язана ним змінна є німою змінною. Тобто, її можна замінити на будь-яку іншу змінну, яка не зустрічається в області дії квантора, при цьому значення істинності ппф не зміниться. Метою перейменування змінних є забезпечення того, щоб кожен квантор мав свою, тільки йому властиву німу змінну.

Наприклад, замість

$$\forall X\{P(X)\to\exists X Q(X)\}$$

треба записати

$$\forall X\{P(X)\to\exists Y Q(Y)\}.$$

Для нашого прикладу отримаємо:

$$\forall X\{\neg P(X)\vee\{\forall Y\{\neg P(Y)\vee P(f(X,Y))\}\wedge\exists W\{Q(X,W)\wedge\neg P(W)\}\}\}.$$

4. ***Вилучення кванторів існування.***

Здійснюється з метою подальшого спрощення логічних формул. Оскільки квантори відображають семантику речень, під час їх вилучення необхідно використовувати спеціальні процедури, які дозволяють зберігати цю семантику. Процедури такого роду вводяться через поняття сколемівської функції.

Для описання сколемівської функції розглянемо такий приклад:

$\forall X\exists Y$ ЛЮБИТЬ (X, Y) – «для кожного X існує такий Y, що X любить Y».

Тобто, якщо взяти будь-який окремий X, то для нього завжди існує такий Y, який задовольняє функцію

ЛЮБИТЬ (X, Y),

або Y = ЛЮБИМИЙ (X).

Отже, відношення між *X* і *Y* може розглядатись у даному випадку як своєрідне функціональне відношення в тому плані, що якщо задано *X*, то існує і відповідний йому *Y*. Тоді *Y* можна замінити на функцію *f(X)*, яка описує дане відношення. Оскільки функція *f(X)* виникла з тих обставин, що *у* відображено квантором існування, то під час підстановки *f(X)* на місце *Y* даний факт буде тим самим відображено і сам по собі квантор існування вже стане не потрібним.

Таким чином, початкову ппф можна переписати у вигляді

$$\forall X\ ЛЮБИТЬ\ (X, f(X)),$$

або що те ж саме, у вигляді

$$\forall X\ ЛЮБИТЬ\ (X, ЛЮБИМИЙ\ (X)).$$

Така функція носить назву сколемівської. Іншим прикладом сколемівської функції може бути такий:

$$\forall Y \exists X\ БІЛЬШЕ\ (X, Y).$$

В даному випадку можна ввести функцію

$$X = ПЕРЕВИЩУЄ\ (Y).$$

Тоді отримаємо

$$\forall Y\ БІЛЬШЕ\ (ПЕРЕВИЩУЄ\ (Y), Y).$$

З наведених прикладів можна побачити, що семантика перетворених формул не порушується.

Загальне правило виключення квантора існування з ппф полягає в заміні кожного входження змінної, яка відноситься до квантора існування, на сколемівську функцію, аргументи якої є змінними зв'язаними кванторами загальності, область дії яких вміщує область дії квантора існування, що вилучається. Таким чином для ппф

$$\forall X \exists Y \forall Z\ F(X, Y, Z)$$

сколемівська функція прийме вигляд

$$\forall X \forall Z\ F(X, f(X), Z),$$

а для ппф

$$\forall X \forall Y \exists Z\ F(X,Y,Z)$$

сколемівська функція прийме вигляд

$$\forall X\ \forall Y\ F(X, Y, f(X,Y)).$$

Якщо квантор існування, що вилучається, не належить області дії жодного з кванторів загальності, то сколемівська функція не містить аргументів, тобто є просто константою. Наприклад,

$$\exists X P(X)$$

перетворюється на

$$P(A),$$

де A – константа, про яку ми знаємо, що вона існує.

Іншим прикладом буде перетворення

$$\exists X \forall Y \text{ ЛЮБИТЬ } (X,Y)$$

на

$$\forall Y \text{ ЛЮБИТЬ } (A,Y),$$

або

$$\text{ЛЮБИТЬ } (A,B).$$

Для нашого прикладу отримаємо:

$$\forall X\{\neg P(X)\vee\{\forall Y\{\neg P(Y)\vee P(f(X,Y))\}\wedge\{Q(X,G(X))\wedge\neg P(G(X))\}\}\}.$$

5. ***Перетворення до випередженої (префіксної) форми.***

На цьому етапі вже не залишилося кванторів існування, а кожний квантор загальності має свою власну змінну. Тепер всі квантори загальності можна перенести на початок формули і вважати, що область дії кожного з них розповсюджується на всю ппф. Це і буде префіксна форма, що складається з двох частин: послідовності кванторів, яку називають префіксом, та наступної безкванторної формули, яку називають матрицею.

В нашому прикладі випереджена форма ппф набуде такого вигляду:

$$\forall X\forall Y\{\neg P(X)\vee\{\{\neg P(Y)\vee P(f(X,Y))\}\wedge\{Q(X,G(X))\wedge\neg P(G(X))\}\}\}.$$

6. ***Зведення матриці до клаузальної нормальної форми (КНФ).***

Будь-яка матриця може бути записана як кон'юнкція скінченної множини диз'юнктивних літералів (кажуть, що така матриця знаходиться в КНФ). Прикладом такої матриці може служити:

$$\{P(X)\vee Q(X,Y)\}\wedge\{P(U)\vee\neg R(Y)\}\wedge Q(X,Y).$$

Будь-яку матрицю можна звести до КНФ багатократним застосуванням дистрибутивного правила, тобто заміною виразів вигляду

$$X1\vee(X2\wedge X3)$$

на

$$(X1\vee X2)\wedge(X1\vee X3).$$

Для нашого прикладу отримаємо:

$$\forall X\forall Y\{\{\neg P(X)\vee\neg P(Y)\vee P(f(X,Y))\}\wedge\{\neg P(X)\vee Q(X,G(X))\}\wedge\{\neg P(X)\vee\neg P(G(X))\}\}.$$

7. ***Вилучення кванторів загальності.***

Оскільки всі змінні у ппф тепер є зв'язаними будь-яким квантором загальності, а порядок розташування кванторів загальності не має значення, можна ці квантори не показувати в явному вигляді, за замовчуванням, вважаючи, що всі змінні в матриці відносяться до кванторів загальності.

Тепер від нашої формули залишається лише матриця, подана у вигляді КНФ:

$$\{\{\neg P(X)\vee\neg P(Y)\vee P(f(X,Y))\}\wedge\{\neg P(X)\vee Q(X,G(X))\}\wedge\{\neg P(X)\vee\neg P(G(X))\}\}.$$

Тобто, початкова ппф вже подана у вигляді нормальної форми. Але, ще більш спростити запис ппф дозволяє використання так званої клаузальної нормальної форми, зведення до якої здійснюється шляхом виконання ще двох додаткових кроків.

8. ***Вилучення символів кон'юнкції.***

Явної присутності символів кон'юнкції можна уникнути, використавши заміну виразів вигляду

$$\{X1\wedge X2\}$$

двома ппф X1 та X2

$$\{X1;X2\}.$$

Висновком багаторазового використання таких замін буде деяка множина ппф, кожна з яких є диз'юнкцією літералів. Будь-яка ппф, яка складається виключно з диз'юнкції літералів, називається реченням. У нашому прикладі ппф перетворюється на таку множину речень:

$$\{\neg P(X)\vee\neg P(Y)\vee P(f(X,Y));\ \neg P(X)\vee Q(X,G(X));\ \neg P(X)\vee\neg P(G(X))\}.$$

Отримана таким чином множина речень називається клаузальною множиною.

9. ***Переіменування змінних.***

Кожне речення клаузальної множини є незалежно заданою ппф. Тому навіть якщо різні речення використовують однакові символи змінних, будь-яка взаємозалежність між ними відсутня. У зв'язку з цим, для зручності подальшого використання символи змінних клаузальної множини змінюють так, щоб вони не з'являлися більше, ніж в одному реченні.

Для нашого прикладу маємо остаточно:

$$\neg P(X1)\vee\neg P(Y)\vee P(f(X1,Y));$$
$$\neg P(X2)\vee Q(X2,G(X2));$$
$$\neg P(X3)\vee\neg P(G(X3)).$$

Таким чином, для використання автоматичної процедури логічного виведення необхідно початкову множину ппф звести до клаузальної нормальної форми.

7.2 Речення Хорна

Розглянуті клаузальні формули (речення) являють собою декілька предикатів та заперечень предикатів (тобто літералів), поєднаних зв'язкою диз'юнкції. Тобто, перетворення до клаузальної форми відповідає заміні описання логічної формули найбільш загального вигляду на групу формул, кожна з яких має такий вигляд:

$$\neg F_1 \vee \neg F_2 \vee \ldots \vee \neg F_n \vee G_1 \vee G_2 \vee \ldots \vee G_m.$$

При цьому, в залежності від значень *n* і *m*, такі формули можна розділити на декілька типів:

1. ***Тип 1; n=0, m =1.***

Таке речення подає одиничний предикат і може буде записане у такому вигляді:

$$\rightarrow F(t_1, t_2, \ldots, t_k),$$

де t_i – є термами.

Якщо всі терми є константами, речення описує деякий факт. Якщо терми містять змінні, то вони задають загальнозначиме подання, що висловлюється для групи фактів.

2. ***Тип 2; n≥1, m =0.***

Такий тип можна задати у вигляді:

$$\neg F_1 \vee \neg F_2 \vee \ldots \vee \neg F_n \rightarrow,$$

або, що теж саме, у вигляді

$$\neg(F_1 \wedge F_2 \wedge \ldots \wedge F_n) \rightarrow.$$

Такий тип використовується для описання питань, оскільки в процедурі автоматичного доведення використовується так званий метод спростування, в якому створюється заперечення запитання і доводиться, що таке заперечення не виконується.

3. ***Тип 3; n≥1, m =1.***

Відповідає поданню знань у формі ЯКЩО... – ТО...:

$$F_1 \wedge F_2 \wedge \ldots \wedge F_n \rightarrow G_m.$$

4. ***Тип 4; n=0, m >1.***

Цей тип речень має вигляд:

$$\rightarrow G_1 \vee G_2 \vee \ldots \vee G_m$$

і використовується для подання нечітких фактів, нечіткість яких виникає внаслідок неповноти інформації в тому сенсі, що неможливо визначити, які саме з предикатів G_i є істинними.

5. ***Тип 5; n≥1, m>1.***

Найбільш загальний тип, до якого, наприклад, відноситься подання вигляду:

$$БАТЬКИ (X, Y) \rightarrow БАТЬКО (X, Y) \vee МАТИ (X, Y).$$

Речення типу 1-3 називають реченнями Хорна. Системи, які обмежені реченнями Хорна, не дозволяють використовувати подання вигляду 4 і 5, але вони порівняно прості в реалізації і тому знайшли широке використання, зокрема в реалізації мови логічного програмування ПРОЛОГ.

7.3 Контрольні запитання

1. Що таке літерал, диз'юнкт, пустий диз'юнкт?
2. Наведіть формули де Моргана.
3. Що таке сколемівська функція і для чого вона використовується?
4. Які нормальні форми подання логічних формул Ви знаєте?
5. Що таке речення Хорна, де вони використовуються і які переваги дає їх використання?
6. Зведіть до префіксної нормальної форми такі речення:
   А) $(\forall X)\ F(X) \rightarrow (\forall Y)\ G(Y)$;
   б) $(\exists X)\ F(X) \rightarrow (\exists X)\ G(X)$;
   в) $(\forall X)\ F(X) \rightarrow (\exists Y)\ G(Y)$;
   г) $(\exists X)\ F(X) \rightarrow (\forall Y)\ G(Y)$;
   д) $(\forall X)\ P(X,Y) \rightarrow (\exists Z)\ [P(X,Z) \vee (\forall U)(Q(U) \rightarrow P(Z,Z))]$.
7. Зведіть до сколемівської нормальної форми:
   а) $(\exists X)[P(X)\&(\forall Y)(S(Y) \rightarrow T(X,Y))]$;
   б) $(\forall X)[Q(X) \rightarrow (\exists Y)(\forall U)(R(X,Y)\&S(Y,U))]$;
   в) $(\forall X)(\forall Y)(\exists Z)(\forall U)(\exists V)[L(X,Y,Z)\&M(Z,U,V)]$;
   г) $(\forall X)[(\forall Y)P(X,Y) \rightarrow (\exists Z)(Q(X,Z)\&R(Z))]$;
   д) $(\forall X)[(\exists Y)P(X,Y) \rightarrow (\forall Z)(Q(X,Z)\vee P(Z))]$;
   е) $(\forall X)[(\exists Y)P(X,Y) \leftrightarrow (\exists Z)Q(X,Z)]$.

8 ФОРМАЛІЗАЦІЯ ПРОЦЕСУ ЛОГІЧНОГО ВИВЕДЕННЯ

8.1 Принцип резолюції для логіки висловлень

Процес доведення в логіці демонструє, що деяка ппф B є теоремою на заданій множині аксіом $A_1, A_2, \ldots, A_n$, тобто логічно випливає з цих аксіом. При цьому треба довести, що ппф F вигляду

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B$$

є тавтологією, тобто приймає значення «істина» для будь-якої інтерпретації.

Однак замість цього достатньо довести, що заперечення цієї формули

$$\neg (A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B)$$

є суперечним, тобто не задовольняється в жодній інтерпретації.

Нехай внаслідок перетворення останньої формули до нормальної форми отримано деяку множину диз'юнктів S. Тоді вихідна формула F є загальнозначимою в тому і тільки в тому випадку, якщо S – суперечна.

Найбільш поширеним методом перевірки множини S на суперечність є метод резолюцій, який базується на таких положеннях:

- якщо S містить пустий диз'юнкт – то вона суперечна;
- якщо S не містить пустого диз'юнкта то перевіряється, чи може бути виведено пустий диз'юнкт з S.

Дійсно, сама S є кон'юнкцією диз'юнктів

$$S = C_1 \wedge C_2 \wedge \ldots \wedge C_n,$$

Тобто, умовою її істинності є умова істинності всіх диз'юнктів, а умовою хибності – хибність хоча б одного з них C_i (що в загальному випадку простіше перевірити).

Однак диз'юнкт C_i можна подати у вигляді

$$C_i = P_{i1} \vee P_{i2} \vee \ldots \vee P_{im},$$

де P_{im} – літерал (атомарний предикат або його заперечення).

Тому умовою того, що в деякій інтерпретації Cin буде хибним є та, що множина {Pi1,Pi2,...,Pim } буде пустою. Якщо це не так, то серед усіх можливих інтерпретацій знайдеться така, що деякий з літералів Pi1,Pi2,...,Pim або всі вони стануть істинними, і тому Ci, в загальному випадку, не буде хибним.

Отже, якщо S містить пусте речення, то формула

$$\neg (A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow B)$$

є хибною, тобто вихідна формула є істинною, а це означає, що твердження B логічно випливає з множини тверджень $A_1, A_2, \ldots, A_n$.

Розглянемо спочатку застосування методу резолюцій до логіки висловлень. Припустимо, що у множині речень присутні додаткові літерали, тобто такі літерали, які відрізняються тільки символом заперечення:

$$C_i = L_{i1} \vee L_{i2} \vee \ldots \vee L_{imi} \vee L;$$

$$C_j = L_{j1} \vee L_{j2} \vee \ldots \vee L_{jmj} \vee \neg L.$$

Вилучимо з цих речень додаткові літерали і подамо у диз'юнктивній формі ту їх частину, що залишилась (таке подання називається резольвентою C_1 і C_2):

$$C=L_{i1}\vee L_{i2}\vee\ldots\vee L_{imi}\vee L_{j1}\vee L_{j2}\vee\ldots\vee L_{jmj}=(C_1-L)\vee(C_2-\neg L).$$

Не важко помітити, що C є логічним висновком з C_1 і C_2. Отже, додання C до множини S не впливає на істинність або хибність S. Якщо ж виконуються умови

$$C_1=L;$$
$$C_2=\neg L,$$

то C пусте.

Той факт, що «C є логічним висновком з S і C – пусте», вказує на хибність початкової логічної формули.

Простий приклад доведення методом резолюцій наведено на рис. 12, а. На рис. 12, б наведено ілюстрацію такого дедуктивного виведення з використанням деревоподібної структури, яка називається дедуктивним деревом, або деревом виведення.

В логічному програмуванні як резольвенти, тобто початкові речення до яких застосовується правило виведення, що називається резолюцією, використовують тільки речення Хорна, тобто речення трьох типів:

заперечення: $\neg(A_1,A_2,\ldots,A_n)$,
факт: A,
імплікація: $A\leftarrow B_1,B_2,\ldots,B_m$,

де $A,A_1,\ldots, A_n, B_1,B_2,\ldots,B_m$ – довільні предикати.

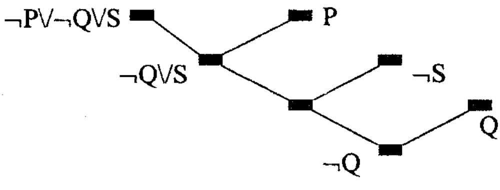
S: $C_1\wedge C_2\wedge C_3\wedge C_4\wedge C_5$

(1) C1: $\neg P\vee\neg Q\vee R$
(2) C2: $\neg P\vee\neg Q\vee S$
(3) C3: P
(4) C4: $\neg S$
(5) C5:Q

Резолюцією отримуємо:

(6) C6: $Q\vee S$ – резольв. (2) і (3)
(7) C7: $\neg Q$ – резольв. (4) і (6)
(8) C8: – резольв. (5) і (6)

$\square$ – пусте речення



а) б)

Рисунок 12 – Доведення теорем методом резолюцій:
а) алгоритм доведення; б) дерево виведення

Розглянемо одну з найпростіших форм резолюції. Нехай є заперечення і імплікація:

$$S_1: \neg A;$$
$$S_2: A\leftarrow B,$$

де предикат A з S_1 збігається з консеквентом A з S_2.

Тоді, внаслідок одного використання правила резолюції, яке називається кроком виведення, з S_1 і S_2 буде отримане нове речення:

$$S=\neg B.$$

Тут S_1 і S_2 називають батьківськими реченнями, а S – резольвентою, яка отримується внаслідок застосування резолюції до S_1 і S_2.

В цьому випадку резолюція відповідає стандартному пропозиційному правилу виведення, яке має назву modus tollens, і суть якого можна виразити таким умовиводом:

припускаючи, що не A, і A якщо B,
отримуємо не B.

Наприклад:

A – студент склав залік;
B – студент добре вчився;
A якщо B –студент добре вчився, то студент складе залік.

Розглянемо ще більш простий випадок, коли S_1 є запереченням, а S_2 – фактом:

$$S_1: A;$$
$$S_2: \neg A.$$

Застосувавши до батьківських речень правило резолюції, як резольвенту отримуємо пусте речення S:

$$\square$$

де символ $\square$ – позначає суперечність. Таким чином, в даному випадку резолюцією є таке міркування: припускаючи, що не A і A, виводимо суперечність.

В реальних випадках мають справу з більш складними реченнями в яких, наприклад, заперечення можуть містити декілька предикатів, а імплікації – декілька антецедентів:

$$S_1: \neg(A_1,A_2,\ldots,A_n);$$
$$S_2: A_k \leftarrow B_1,B_2,\ldots,B_m,$$

де $n \geq k \geq 1$.

Тут деякий предикат A_k із заперечення S_1 збігається з консеквентом з імплікації S_2. В такій ситуації крок виведення замінює A_k в S_1 на антецедент з S_2 і як резольвенту отримуємо заперечення

$$S: \neg(A_1,A_2,\ldots,A_{k-1}, B_1,B_2,\ldots,B_m, A_{k+1},\ldots,A_m)$$

Якщо ж S_1 має той же вигляд, що й у вищенаведеному прикладі, а S_2 є просто фактом

$$S_2: A_k,$$

причому A_k є одним з предикатів в S_1, крок виведення просто викреслює A_k з S_1 і результатом є резольвента

$$S: \neg(A_1, A_2, ..., A_{k-1}, A_{k+1}, ..., A_m).$$

8.2 Уніфікація

Використовуючи метод резолюції в логіці предикатів, алгоритм доведення трохи ускладнюється за рахунок того, що на відміну від логіки висловлень тут всередині атомарних формул (предикатів) можуть знаходитися змінні. У зв'язку з цим, перед застосуванням алгоритму виведення, необхідно виконати деякі підстановки в змінні, і за рахунок таких підстановок провести уніфікацію батьківських речень.

Розглянемо два предикати L(X) і L(A). Припустимо, що X – змінна, а A – константа. В таких предикатах предикатні символи є однаковими, але про самі предикати цього сказати не можна. Проте за рахунок підстановки значення A на місце змінної X такі предикати можна зробити однаковими. Саме така підстановка і отримала назву уніфікації. Метою уніфікації є забезпечення можливості застосування алгоритму автоматичного доведення за допомогою методу резолюцій для логіки предикатів.

Наприклад, у формулах

$$C_i = L_{i1} \vee L_{i2} \vee ... \vee L_{imi} \vee L(X),$$
$$C_j = L_{j1} \vee L_{j2} \vee ... \vee L_{jmj} \vee \neg L(A)$$

літерали L(X) і $\neg$L(A) не знаходяться у додаткових відношеннях. Але виконавши підстановку A замість X отримаємо, відповідно, літерали L(A) і $\neg$L(A), які вже знаходяться в додаткових відношеннях. Але під час здійснення підстановок треба враховувати певні обмеження.

Підстановку терму *t* на місце змінної X прийнято записувати у вигляді

$$\{t/X\}.$$

Оскільки в одній ппф може бути декілька змінних, можна здійснювати і більше ніж одну підстановку. Тому підстановки записують у вигляді множини впорядкованих пар:

$$Q = \{t_1/X_1, t_2/X_2, ..., t_n/X_n\}.$$

Умовами за яких передбачається підстановка є такі:

- X_i є змінною, а t_i – термом (константа, змінна або функціональний символ), який відрізняється від X_i;
- для будь-якої пари елементів з групи підстановок, наприклад, $\{t_1/X_1, t_2/X_2\}$ в правих частинах символів '/' однакові змінні відсутні.

Таким чином, підстановкою Θ називається будь-яка множина присвоювань вигляду $X:=t$, де X – змінна, а t – терм, причому кожній змінній присвоюється не більше одного значення. Застосування Θ до деякого довільного виразу L, наприклад до предиката, полягає в заміні змінних з L на терми, які згідно з Θ надаються цим змінним. Змінні з L, які не згадуються в Θ, залишаються без змін, а присвоювання Θ змінним, що не входять до складу L, не виконуються. Результат застосування Θ до L позначається як L_Θ і називається підстановним прикладом (варіантом, випадком) L. Якщо результатом застосування Θ до L є лише заміна одних змінних на інші, такий підстановний приклад називається алфавітним. Якщо результатом застосування Θ до L стане заміна всіх змінних константами, то такий підстановний приклад називається константним, або атомом, оскільки ні в одному з термів літерала не будуть присутні змінні.

Якщо застосування Θ до двох різних предикатів L_1 і L_2 дає однакові підстановні приклади, то вираз $L_1=\Theta=L_2$ називається загальним прикладом L_1 і L_2, а підстановка Θ називається в даному випадку уніфікатором, оскільки її застосування стискає множину до одного елемента. Якщо така підстановка існує, то кажуть, що множина виразів $\{L_1,L_2\}$ є такою, що уніфікується, при цьому уніфікатором є підстановка Θ. В попередньому прикладі множина $\{L(X),L(A)\}$ є такою, що уніфікується, а уніфікатором є підстановка $\{A/X\}$.

Для однієї групи виразів не обов'язково існує лише один уніфікатор. Наприклад, для групи виразів

$$\{L(X,Y), L(Z,f(X))\}$$

підстановка

$$\Theta=X/Z, f(X)/Y\}$$

є уніфікатором, але уніфікатором є і підстановка

$$\Theta=\{A/X, A/Z, f(A)/Y\}.$$

Помітимо, що в першому випадку ми отримали алфавітний, а в другому – константний підстановні приклади. В таких випадках виникає проблема, яку ж саме підстановку краще вибрати як уніфікатор. Для з'ясування цього відмітимо, що операцію підстановки можна проводити не за одним разом, а розділити її на декілька етапів. Її можна також розділити по групах змінних, провівши, наприклад, підстановку

$$\{t_1/X_1, t_2/X_2, t_3/X_3, t_4/X_4\}$$

спочатку для

$$\{t_1/X_1, t_2/X_2\},$$

а потім для

$$\{t_3/X_3, t_4/X_4\}.$$

Можна також підстановку вигляду

$$\{A/X\}$$

розбити на дві підстановки

$$\{U/X\}$$

і

$$\{A/U\}$$

(вважається, що X і U змінні).

Результат послідовного виконання двох підстановок θ і λ також називається підстановкою і позначається як композиція θ λ.

Композицією $\theta \circ \lambda$ двох підстановок θ і λ називається результат застосування λ до термів підстановки θ з наступним доданням тих пар з λ, які містять змінні, що не зустрічалися серед змінних з θ.

Наприклад, для

$$\theta=\{G(X,Y)/Z\};$$
$$\lambda=\{A/X, B/Y, C/W, D/Z\}$$

отримаємо

$$\theta \circ \lambda= \{G(A,B)/Z, A/X, B/Y, C/W\}.$$

Якщо існує декілька уніфікаторів, то серед них завжди знайдеться така підстановка θ, що будь-які інші уніфікатори будуть підстановками, які можна виразити у вигляді композиції $\theta \circ \lambda$, тобто будуть складеними уніфікаторами, що містять дану підстановку θ. Така підстановка називається найбільш загальним (найбільш простим) уніфікатором (НЗУ) і завжди існує з точністю до алфавітних варіантів.

Відмітимо що внаслідок підстановок змінні заміщуються константами, що певною мірою обмежує описувальну потужність ппф. Для уніфікації двох різних предикатів необхідна така підстановка, за допомогою якої вираз, що має більшу описувальну потужність, узгоджується із виразом, що має меншу описувальну потужність. Однак, оскільки кожна підстановка обмежує цю описувальну потужність, то кількість підстановок повинна бути мінімально необхідною. НЗУ якраз і є уніфікатором із мінімальними обмеженнями такого типу.

Метод пошуку НЗУ називається алгоритмом уніфікації і є відносно простим. Він базується на почерговому порівнянні відповідних аргументів предикатів і виписуванні тих присвоювань термів змінним, які зробили б ці аргументи однаковими. Припустимо, наприклад, що у нас є такі літерали:

$$L(A, t, f(Z)), L(A,X, Z).$$

Оскільки перші аргументи літералів збігаються, переходимо до розгляду других аргументів. Враховуючи, що:

(1) X є змінною

X не міститься в t, створюємо перший елемент множини підстановок $\{t/X\}$. Під час порівняння третіх аргументів *Z* і *f(Z)* умова (1) виконується, але умова (2) не виконується і тому підстановка не є можливою. Тобто, в даному прикладі уніфікатор відсутній і розглянуті літерали не є такими, що уніфікуються.

Речення можна розглядати як множини літералів {Li}, які, в загальному випадку, також можна уніфікувати за допомогою НЗУ λ. При цьому отримане речення $\{Li\}_{\lambda}$ називають фактором вихідного речення {Li}. Наприклад, для речення:

$$P(f(X)) \vee P(X) \vee Q(A,f(u)) \vee Q(X,f(B)) \vee Q(Z,W)$$

факторами будуть такі речення:

$$P(f(Z)) \vee P(Z) \vee Q(A,f(U)) \vee Q(Z,f(B)); \qquad \theta=\{Z/X,\ f(B)/W\}$$

$$P(f(A)) \vee P(A) \vee Q(A,f(B)); \qquad \theta=\{A/X,\ B/U,\ A/Z,\ B/W\}$$

(2)

На рисунку 13 наведено ще декілька простих прикладів уніфікації.

Батьківські речення	Уніфікатор	Загальний приклад
S1: p(5) S2: p(5)	Пуста множина	S: p(5)
S1: p(X) S2: p(5)	θ={5/X}	S: p(5)
S1: p(X) S2: p(Y)	θ={Y/X}	S: p(Y)
S1: p(X,X) S2: p(5,Y)	θ={5/Y, 5/X}	S: p{5,5}
S1: p(f(X),f(5),X) S2: p(Z,f(Y),Y)	θ={f(5)/Z, 5/Y, 5/X}	S: p(f(5),f(5),5)
S1: ¬p(5,Y) S2: g(X)→p(X,X)	θ={5/X, 5/Y}	S: ¬g(5)
S1: L(f(X),Y,g(Y)) S2: L(f(X),Z,g(X))	θ= {X/Y,X/Z}[θ={Z/Y,X/Z}]	S: L(f(X),X,g(X))
S1: (f(X,g(A,Y)),g(A,Y)) S2: L(f(X,Z),Z)	θ={g(A,Y)/Z}	S: L(f(X,g(A,Y)),g(A,Y)

Рисунок 13 – Приклади уніфікації речень

У першому факторі уніфіковані тільки два останніх літерали Q, а у другому – всі три. В цьому реченні літерали P не уніфікуються. Взагалі в реченні може бути більше ніж один фактор, але їх загальна кількість завжди скінченна.

Наведемо неформальну рекурсивну процедуру уніфікації множини з двох виразів L1 і L2:

RecurSive proceDure UNIFY(L1,L2)

1) **if** L1 або L2 є атом, то поміняти місцями аргументи L1 і L2 (якщо необхідно) так, щоб атомом був L1 і **Do:**
2) **Begin**
3) **if** L1 ідентична L2, **return NIL**
4) **if** L1 є змінною **Do:**
5) **Begin**
6) **if** L1 міститься в L2, **return FAIL**
7) **return** {L2/L1}
8) **enD**
9) **if** L2 є змінною **return** {L1/L2}
10) **return FAIL**
11) **end**
12) H1 ← перший елемент з L1, T1 – залишок L1
13) H2 ← перший елемент з L2, T2 – залишок L2
14) Z1 ← UNIFY(H1,H2)
15) **if** Z1=FAIL, **return** FAIL
16) G1 ← результат застосування Z1 до T1
17) G2 ← результат застосування Z1 до T2
18) Z2 ← UNIFY(G1,G2)
19) **if** Z2=FAIL, **return** FAIL
20) **return** Z1 Z2 (композиція)

Процедура UNIFY завжди знаходить НЗУ для множини речень, що уніфікуються, і повідомляє про невдачу, якщо речення не уніфікуються.

8.3 Система спростування на основі резолюції

Метод резолюцій у логіці предикатів відрізняється від методу резолюцій у логіці висловлень тільки тим, що додаткові літерали можуть створюватись тут за допомогою операції підстановки.

Розглянемо речення $\{L_i\}$ і $\{M_i\}$, змінні одного з яких не зустрічаються в іншому. Нехай для підмножин $\{l_i\} \subseteq \{L_i\}$ і $\{\neg m_i\} \subseteq \{M_i\}$ існує найбільш загальний уніфікатор λ. Тоді кажуть, що два речення $\{L_i\}$ і $\{M_i\}$ розв'язуються, а їх резольвентою називають нове речення

$$[\{L_i\}-\{l_i\}]_\lambda \cup [\{M_i\}-\{m_i\}]_\lambda.$$

При розв'язанні двох речень може бути утворена більше ніж одна резольвента, але число резольвент завжди скінченне.

Наприклад, розглянемо речення:

$$\{L_i\}=\{P(X,f(A)) \vee P(X,f(Y)) \vee Q(Y)\};$$
$$\{M_i\}=\{\neg P(Z,f(A)) \vee \neg Q(Z)\}.$$

Обравши

$$\{l_i\}=\{P(X,f(A))\},$$
$$\{m_i\}=\{\neg P(Z,f(A))\},$$

отримаємо резольвенту:

$$P(Z,f(Y)\vee \neg Q(Z)\vee Q(Y).$$

Якщо ж вибрати

$$\{l_i\}=\{P(X,f(A)),\ P(X,f(Y))\}$$
$$\{m_i\}=\{\neg P(Z,f(A))\}$$

отримаємо резольвенту

$$\neg Q(Z)\vee Q(A).$$

Всього ж для цих двох речень є чотири резольвенти, три з яких отримуються розв'язанням на P і одна – розв'язанням на Q.

Резолюція є загальним правилом виведення. Наприклад, резольвента

$$\neg P(X)\vee Q(X)$$

двох речень

$$\neg P(X)\vee R(X),$$
$$\neg R(X)\vee Q(X),$$

відповідає такому ланцюжку міркувань:

$(\forall X)\{P(X)\rightarrow R(X)\}$ – все, що має властивість P має і властивість R;

$(\forall X)\{R(X)\rightarrow Q(X)\}$ – все, що має властивість R має і властивість Q;

отже,

$(\forall X)\{P(X)\rightarrow Q(X)\}$ – все, що має властивість P має і властивість Q.

Основані на резолюції системи призначені для здійснення автоматичного доведення теорем, яке основується на побудові суперечності або спростування.

В типовій задачі на доведення теорем є множина S правильно побудованих формул, виходячи з якої треба довести цільову ппф W. Для цього беруть заперечення цільової ппф і додають його до вихідної множини S, отримуючи множину S′

$$S' = S \cup \neg W.$$

Доповнена таким чином множина S′ далі перетворюється на множину речень, після чого використовується резолюція з метою спроби виведення суперечності, яка подається пустим реченням NIL. Розглянемо простий приклад такого процесу.

Нехай висловлені такі твердження:

- хто має багато грошей, той багатий;
- студенти не багаті;
- деякі студенти мають великі знання.

  Треба довести твердження:

- деякі з тих, хто має великі знання не мають грошей.

Формалізуємо дані речення, для чого введемо такі предикатні символи:

Г – має багато грошей;

Б – багатий;

С – студенти;

З – мають великі знання.

Після формалізації вихідні твердження набудуть такого вигляду:

1. $(\forall X)[Г() \rightarrow Б(X)]$
2. $(\forall X)[С(X) \rightarrow \neg Б(X)]$
3. $(\exists X)[С(X) \land З(X)]$
4. $(\exists X)[З(X \land \neg Г(X)]$

Наступним етапом є зведення отриманих ппф 1 – 4 до клаузальної нормальної форми.

До ппф 1 і 2 достатньо застосувати кроки 1 і 7 процедури нормалізації (вилучення символів імплікації та кванторів загальності, відповідно).

До ппф 3 слід застосувати кроки 4 і 8 процедури нормалізації (вилучення кванторів існування та символів кон'юнкції, відповідно).

Внаслідок нормалізації множина речень, що відповідають ппф 1 – 3, набуде такого вигляду:

1. $\neg Г(X) \lor Б(X)$.
2. $\neg С(Y) \lor \neg Б(Y)$.

3, а. $С(A)$.

3, б. $З(A)$.

де змінні розділені, а A є сколемівською константою.

Щодо ппф 4, яку треба довести, то перед нормалізацією необхідно взяти її заперечення:

$$\neg\{(\exists X)[З(X) \land \neg Г(X)]\} = (\forall X) \neg \{[З(X) \land \neg Г(X)]\} = (\forall X)[\neg З(X) \lor Г(X)]\}.$$

Нормалізація даної ппф зводиться лише до вилучення квантора загальності, після чого вона набуде такого вигляду:

4. $\neg З(Z) \lor Г(Z)$.

Тепер виконаємо процедуру доведення спростуванням. Для цього будемо будувати резольвенти для речень 1 – 4, додаючи їх до початкової множини, до тих пір, доки не буде отримано пусту множину.

Одне з можливих дерев виведення (а їх може бути декілька, в залежності від того, які саме резольвенти і в якому саме порядку будуть будуватися) зображено на рисунку 14.

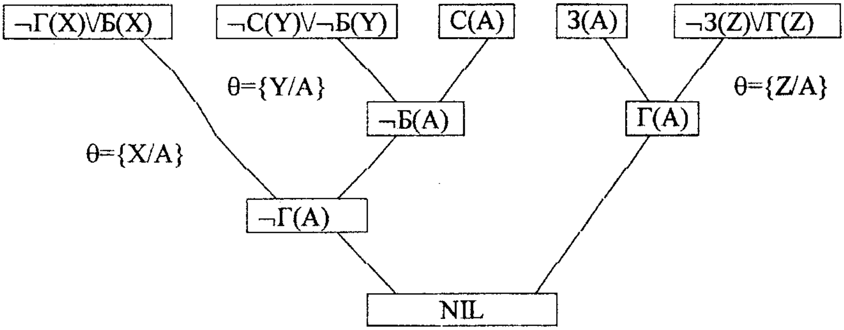


Рисунок 14 – Дерево виведення для задачі про студентів

У загальному випадку одне й те ж саме речення може кілька разів використовуватись як батьківське, створюючи резольвенти з різними реченнями.

Система спростування дозволяє використовувати необоротний режим керування. Це не означає, що кожна отримана резолюція лежить на шляху до мети, частина їх, як правило, виявляється непотрібною. Але «надлишкові» резолюції лише затримують процес, не заводячи його в глухий кут.

Знов звернемось до хорнівських диз'юнктів. Їх розділяють на два види:

– із заголовком (має форму літерала без заперечення або форму диз'юнкта, що містить як літерали без заперечення, так і один або декілька літералів із запереченням):

A:– A

A:–B,C,D ¬B ∨¬C ∨¬D ∨ A

– без заголовка (не має літералів без заперечення):

:–A ¬A

:–B,C,D ¬B ∨¬C ∨¬D

Зауважимо, що під час використання методу резолюцій для двох хорнівських диз'юнктів із заголовком завжди буде знову отримано хорнівський диз'юнкт із заголовком. Але пустий диз'юнкт не має заголовка. Тобто, для отримання можливості виведення пустого диз'юнкта необхідно мати хоча б один диз'юнкт без заголовка.

Якщо декілька з вихідних диз'юнктів не мають заголовків, то доведення кожного нового диз'юнкта методом резолюцій перетворюється на доведення, в якому використовується не більше як один диз'юнкт без заголовка. Якщо пустий диз'юнкт випливає з даної множини вихідних диз'юнктів, то він випливає і з її підмножини, яка містить кілька диз'юнктів із заголовками і не більше одного диз'юнкта без заголовка. Тобто, будь-яка задача, яка може бути подана за допомогою хорнівських диз'юнктів повинна мати лише один диз'юнкт без заголовка, а усі інші диз'юнкти повинні мати заголовки.

Спрощений варіант методу резолюцій, що обмежений використанням лише хорнівських диз'юнктів, використовується в Пролозі. Принциповою відмінністю алгоритму резолюцій на множині хорнівських диз'юнктів є те, що на кожному кроці деяка літера видаляється з одного диз'юнкта. З цього випливає, що робота алгоритму завжди закінчиться, незалежно від вибраної стратегії. При цьому, якщо S спочатку містить N літер, то цикл буде виконуватись не більше як N разів, тобто, складність алгоритму не перевищить N^2. І буде або отримано пустий диз'юнкт або отримано множину диз'юнктів, в якій жодний з диз'юнктів із заголовком не буде містити заперечення унітарного диз'юнкта без заголовка.

8.4 Отримання відповіді зі спростування, яке основане на резолюції

Часто системи доведення теорем в численні предикатів використовують для формул, які містять змінні, пов'язані кванторами існування. При цьому з'являється можливість знайти конкретні значення зв'язаних змінних, які логічно випливають з множини S. Тобто, не тільки підтвердити існування відношення логічного висновку між ппф і заданою множиною тверджень S, але й знайти конкретні значення змінної, пов'язаної квантором існування.

Розглянемо таку просту задачу:

«Якщо Роман завжди ходить разом з Іваном і якщо Іван знаходиться у школі, то де знаходиться Роман?».

В задачі визначено два факти і задається запитання. При цьому припускається що відповідь на запитання можна вивести з наведених фактів. Вказані факти можна перетворити на таку множину ппф:

1. $(\forall X)[B(Іван, X) \rightarrow B(Роман(X)]$
2. B(Іван, школа)

На запитання

«Де знаходиться Роман?»

можна відповісти якщо спочатку довести, що ппф

3. $(\exists X)B(\text{Роман}, X)$

випливає з S, а потім знайти конкретний випадок X, «який існує». Тобто, необхідно перетворити запитання у цільову ппф, що містить квантор існування, таким чином, щоб змінна, зв'язана квантором існування в ппф, давала б відповідь на запитання.

Спростування на основі резолюції можна отримати звичайним шляхом, взявши заперечення ппф, яке необхідно довести.

При цьому, внаслідок формалізації перших двох ппф отримуємо:

$$\neg B(\text{Іван}, Y) \vee B(\text{Роман}, Y);$$

$$B(\text{Іван}, \text{школа}).$$

Заперечення ппф, що треба довести, набуде вигляду

$$\neg(\forall X)[B(\text{Роман}, X)],$$

а після нормалізації буде подане формулою:

$$\neg B(\text{Роман}, X).$$

Дерево спростування для нашого прикладу наведено на рисунку 15.

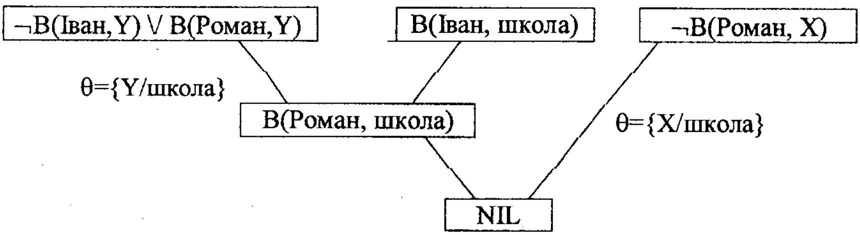


Рисунок 15 – Дерево спростування для задачі про Романа та Івана

Тепер необхідно з дерева спростування знайти відповідь на запитання «Де Роман?».

Це можна здійснити таким чином.

1. Додати до кожного речення, яке виникло із заперечення цільової формули, його власне заперечення. В нашому випадку заперечення

$$\neg B(\text{Роман}, X)$$

буде замінено на тавтологію

$$\neg B(\text{Роман}, X) \vee B(\text{Роман}, X).$$

2. Виконати ті ж самі резолюції, що і в розглянутому дереві спростування, внаслідок чого в кореневій вершині дерева буде побудоване деяке речення.
3. Використати речення з кореневої вершини як відповідь.

Дерево доведення з відповіддю на запитання у кореневій вершині для нашого прикладу наведене на рисунку 16.

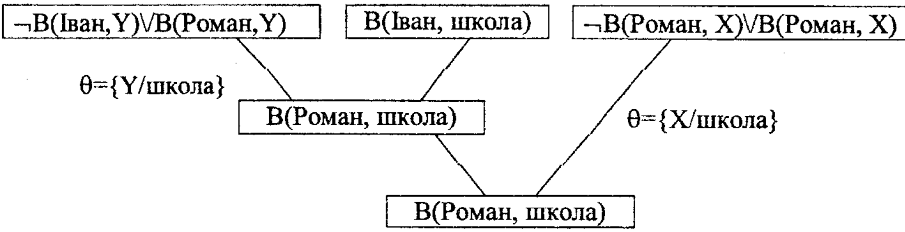


Рисунок 16 – Вилучення відповіді з дерева спростування

З рис. 16 можна побачити, що відповідь має форму, подібну до форми цільової ппф, але на місці змінної отримано константу, яка відноситься до квантора існування цільової формули.

8.5 Контрольні запитання

1. Наведіть основні положення методу резолюцій.
2. Що таке резольвента і за якими правилами вона утворюється?
3. Що означає термін «пусте речення»?
4. Наведіть власний приклад невеличкого дерева виведення.
5. Що таке уніфікація і для чого вона виконується?
6. Сформулюйте правила уніфікації термів.
7. Наведіть умови, за яких припускається здійснення підстановки.
8. Що таке уніфікатор?
9. Сформулюйте правила обчислення композиції підстановок.
10. Охарактеризуйте поняття найбільш загального уніфікатора.
11. Поясніть поняття «фактор речення».
12. Наведіть узагальнений алгоритм виведення методом резолюцій.
13. Доведіть за допомогою методу резолюцій, що формула G є логічним висновком формул $F_1,\ldots,F_n$:
 $F_1=X\&Y\to\neg X\&Z$, $F_2=\neg(X\&\neg Y)\vee Z$, $G=X\to Z$.
14. Запишіть нижченаведені міркування у вигляді послідовності формул логіки висловлень і перевірте методом резолюцій чи логічні вони: «Якщо конгрес відмовляється прийняти нові закони, то страйк не буде закінчено, окрім випадку, якщо він продовжується більше місяця і президент фірми піде у

відставку. Припустимо, що конгрес відмовився діяти і страйк закінчився. Отже, страйк продовжувався більше місяця.

15. Визначте чи уніфікуються такі множини атомарних формул:

А) M={P(A,Y,Y), P(Z,X,f(X))},

б) M={P(X,Y,Z), P(U,h(Y,Y),Y), P(A,B,C)},

в) M={P(A,f(X),g(X,Y)), P(U,Y,g(f(A),h(Y)))}?

ЛІТЕРАТУРА

1. Братко Т. А. Программирование на языке Пролог для искусственного интеллекта / Братко Т. А.; [пер. с англ. Т. А. Братко]. – М. : Мир, 1990. – 560 с.
2. Клоксин У. Программирование на языке Пролог / У. Клоксин, Д. Меллиш ; [пер. с англ. У. Клоксин, Д. Меллиш]. – М. : Мир, 1987. – 336 с.
3. Стобо Д. Ж. Язык программирования Пролог / Стобо Д. Ж.; [пер. з англ. – Д. Ж. Стобо]. – М. : Радио и связь, 1993. – 368 с.
4. Доорс Дж. Пролог – язык программировани будующего / Дж. Доорс, А. Р. Рейблейн, З. Вадера; [пер. з англ. Дж. Доорс, А. Р. Рейблейн, Вадера З.]. – М. : Финансы и статистика, 1990. – 144 с.
5. Стерлинг Л. Искусство программирования на языке Пролог / Л. Стерлинг, Э. Шапиро; [пер. с англ. Л. Стерлинг, Э. Шапиро]. – М. : Мир, 1990. – 235 с.
6. Малпас Дж. Реляционный язык программирования Пролог / Малпас Дж.; [пер. с англ. Дж. Малпас]. – М. : Наука, 1990. – 463 с.
7. Хоггер К. Введение в логическое программирование / Хоггер К.; [пер. с англ. К. Хоггер]. – М. : Мир, 1983. – 384 с.
8. Месюра В. І. Функціональне та логічне програмування. Частина 1. Основи функціонального програмування / Месюра В. І. – Вінниця : ВДТУ, 2001.
9. Адаменко А. И. Логическое програмированние и ViSuAl Prolog / А. И. Адаменко, А. М. Кучуков – Спб. : БХВ – Петербург, 2003.
10. Братко Иван. Алгоритмы искусственного интеллекта на языке PROLOG, 3-е издание / Братко Иван.; [пер. с англ. Иван Братко]. – М. : Издательский дом «Вильямс», 2004.
11. Шнайдер В. А. Основы программирования на языке Пролог: курс лекций: учебн. пособие для студентов вузов, обучающихся по специальностям в обл. информ. технологий / Шнайдер В. А. – М. : Ун–т информ. технологий, 2005. – 176 с.

Навчальне видання

Месюра Володимир Іванович
Лисак Наталія Володимирівна
Суприган Олена Іванівна

МАТЕМАТИЧНІ ОСНОВИ ЛОГІЧНОГО ПРОГРАМУВАННЯ

Навчальний посібник

Редактор В. Дружиніна
Коректор З. Поліщук

Оригінал-макет підготовлено Суприган О. І.

Підписано до друку 20.08.2013 р.
Формат 29,7 х 42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різографічний. Ум. друк. арк. 6,0.
Наклад 75 прим. Зам. № 2013-029.

Вінницький національній технічний університет,
навчально-методичний відділ ВНТУ.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 2201.
Тел. (0432) 59-87-36.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-87-38.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.