31. Анохин, П. К. Принципиальные вопросы общей теории функциональных систем / П. К. Анохин // От моделей поведения к искусственному интеллекту. – 2006. – С. 9-60.
32. Фуфаев, В. В. Ценологическое определение параметров электропотребления, надёжности, монтажа и ремонта электрооборудования предприятий региона [Текст] / В. В. Фуфаев. – М. : Центр системных исследований, 2000. – 320 с.
33. Tapp, B. A., Koiuszko V. Application of Zipfs law to mineral distribution patterns in the northern Australian orogenic provinces. - Proc.11 th Commonw. Mining and Met. Congr. Hong, 1978, London, 1979. – p. 261-268.
34. Семиходский, Г. Е. Прогноз газоносности ДДВ на основе статистических данных [Текст] / Г. Е. Семиходский, Ю. В. Тимошин // Геология нефти и газа. – 1982. – №7. – 8 с.
35. Максимов, В. Н., Концепция выявления стрессовых состояний водных экосистем методом ранговых распределений и экологически допустимые уровни загрязняющих веществ для водоемов р. Элиста [Текст] / В. Н. Максимов, Л. В. Джабруева, Н. Г. Булгаков, А. Т. Терехин // Водные ресурсы. – 1997. – Т. 24, № 1. – С. 79–85.
36. Бурков, В. Н. Теория активных систем: состояние и перспективы [Текст] / В. Н. Бурков, Д. А. Новиков. – М. : Синтег, 1999. – 128 с.
37. Касьянов В. А. Элементы субъективного анализа [Текст] / В. А. Касьянов – К. : НАУ, 2003. – 224 с.
38. Касьянов В. А. Субъективный анализ: Монография [Текст] / В. А. Касьянов. – К. : НАУ, 2007. – 512 с.
39. Саймон, Г. А. Теория принятия решений в экономической теории и науке о поведении [Текст] / Г. А. Саймон // Теория фирмы. – 1995. – С. 54 - 72.
40. Саймон Г. А. Характеристики ограниченной рациональности. Рациональное принятие решений в бизнесе [Текст] / Г. А. Саймон // Мировая экономическая мысль. Сквозь призму веков. Кн.1. – 2004. – Кн.1. – С. 331-339.
41. Пущин С. Л. Ценология – это просто [Текст] / С. Л. Пущин // Ценологические исследования. –2010. – Вып.45. – 68 с.
42. Хайтун С. Д. Негауссовость социальных явлений и эволюция [Текст] / С. Д. Хайтун // Электрификация металлургических предприятий Сибири. – 2005. – Вып.12. – С. 291-300.

УДК 510.6

СИСТЕМИ КОМП'ЮТЕРНОЇ АЛГЕБРИ В МАТЕМАТИЧНІЙ ЛОГІЦІ

Г. І. Бедратюк
Асистент
Кафедра програмної інженерії
Хмельницький національний університет
вул. Інститутська 11, м. Хмельрицький, Україна, 29016
Контактний тел.: (03822) 4-90-43
E-mail: bedratyuk@ukr.net

Дано опис команд пакету logic системи комп'ютерної алгебри Maple. Розглянуто способи розв'язання деяких типових задач математичної логіки в Maple

Ключові слова: математична логіка, Maple

Дано описание команд пакета logic системы компьютерной алгебры Maple. Рассмотрены способы решения некоторых типичных задач математической логики в Maple

Ключевые слова: математическая логика, Maple

A description of the commands for the logic package of the computer algebra system Maple are given. The methods of solving some common mathematical logic problems with Maple are considered

Keywords: mathematical logic, Maple

1. Вступ

Система комп'ютерної алгебри Maple перший реліз якої випущений у 1981 році канадською фірмою Waterloo Maple, Inc., успішно поєднує символьні маніпуляції, обчислювальну математику, потужну графіку та зручну мову програмування. В силу своєї зручності та універсальності система Maple стала незамінним інструментом наукових досліджень для багатьох вчених, інженерів та студентів. Останнім часом спостерігається активне проникнення систем комп'ютерної алгебри в освітній процес оскільки це дає можливість формування принципово нових технологій навчання [1,2,3]. Практично для кожного розділу математики в Maple розроблено окремий спеціалізований пакет команд. В даній статті ми розглянемо основні команди пакету logic який розроблений для розв'язання типових задач двозначної булевої логіки, які зустрічаються в процесі вивчення дисципліни "Математична логіка". Для розуміння роботи потрібно мати початкові навики роботи в Maple для чого достатньо ознайомитися з одним із багатьох доступних посібників, наприклад [4], [5]. Також ми вважатимемо, що читач знайомий із основами математичної логіки, див. [6],[7].

2. Опис пакету logic

Дамо короткий опис тієї частини мови програмування системи Maple та тих елементів цієї програми, які необхідні для вирішення основних завдань математичної логіки.

В середовищі Maple для логічних виразів передбачено спеціальний тип змінних – **boolean**. Довільний вираз має тип **Boolean**, які він має тип **relation**, або тип **logical** або є однією із булевих констант: **true, false**, чи **FAIL**. Тип **relation** утворює арифметичні вирази, до яких входять відношення **`=`**, **`<>`**, **`<`**, or **`<=`**. Вираз має тип logical, якщо до нього входять логічні операції **`and`**, **`or`**, **`xor`**, **`implies`** та **`not`**. Для перевірки типу логічного виразу використовується команда *type*:

> type (1 < 2 and x > 5, boolean);

true

> type (x < 5, relation);

true

> type (a or not b, logical);

true

Для обчислення логічних виразів використовується команда **evalb(B)**, яка повертає можливі значення логічного виразу **B**: **true, false**, and **FAIL**. Якщо обчислення є неможливими, то повертається сам вираз **B**.

> evalb(x>y);

y<x

У цьому прикладі обчислення виявилися неможливими, оскільки змінні *x* та *y* не були визначені попередньо. Тому потрибно використовувати такі команди

> x:=1; y:=2;evalb(x>y);

x:=1

y:=2

false

Після запуску програми Maple на екрані монітора комп'ютера з'являється головне вікно програми, в якому вже автоматично відкрита перша сторінка для вирішення завдань. Вона має назву "Untitled (1)". Система Maple складається з декількох груп основних математичних підпрограм, а також з додаткових пакетів, які підключаються в міру потреби. Для задач математичної логіки потрібен додатковий пакет logic. Для підключення цього пакету потрібно в робочому рядку після символу запрошення вводу команди > набрати командний рядок такого вигляду:

> with(logic);

В пакеті logic реалізовано наступні команди

&and, &iff, &implies, &nand, &nor,&not,&or,&xo
BooleanSimplify, Canonicalize,Contradiction,
Dual,Environment,Equivalent,Export,Implies,
Import,Normalize,Random,Satisfy,Tautology,TruthTable

Зауважимо, що рядок закінчується крапкою з комою - цим символом повинні закінчуватися всі ті рядки виконуваних системою команд, результати, дії яких ми хочемо побачити на екрані. Якщо команду потрібно виконати, але результат виводити на екран не потрібно, то в кінці ставиться двокрапка.

Після набору команди

> with(logic);

потрібно натиснути на клавішу вводу (Enter), яка змушує Maple виконати ту команду, яка розташована в рядку курсора. На екрані виведеться список команд доступних в цьому пакеті.

Приклади запису основних логічних операцій у Maple подано у наступній таблиці

Кон'юнкція	&and
Рівносильність	&iff
Імплікація	&implies
Штрих Шеффера	&nand
Стрілка Пірса	&nor
Заперечення	&not
Диз'юнкція	&or
Виключна диз'юнкція	&xor

Зауважимо, що мова програмування Maple використовує тризначну логіку із логічними константами *True, False, FAIL* та логічними операторами *and, iff, implies, nand, nor, not, or, xor*, таблиці істинності яких дещо відрізняються від таблиць істинності відповідних команд стандартної двозначної булевої логіки прийнятої в пакеті Logic. Взаємний перехід між цими двома логічними системами реалізовується командами Export та Import.

Для прикладу наступна команда

> Export((a &or b) &and c, form=boolean);

(a or b) and c

переводить логічний вираз двозначної логіки пакету Logic у відповідний йому логічний вираз тризначної логіки Maple. Команда Import має аналогічний синтаксис.

Перейдемо до аналізу та поясненню основних команд пакету **logic**.

Команда **BooleanSimplify(B)** повертає мінімальну суму добутків, у які може бути розкладений даний логічний вираз **B**.

Приклад 1. Спростити логічний вираз $(a \vee b) \wedge (\bar{a} \vee c)$

> BooleanSimplify((a &or b) &and (&not a &or c));

(a &and c) &or (b &and &not(a))

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

Команда **Canonicalize** перетворює задану формулу до диз'юнктивної або кон'юнктивної досконалих нормальних форм. Синтаксис та особливості використання цієї команди розглянемо на прикладах.

Приклад 2. Знайти ДДНФ формули $a \wedge (b \vee c)$.

> Canonicalize(a &and (b &or c),{a,b,c}, form=DNF);

(((a&andb)&andc)&or((a&andb)&and &not(c)))&or

((a &and c) &and &not(b))

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

Приклад 3. Знайти ДКФ формули $a \wedge b$.

> Canonicalize(a &and b,{a,b}, form=CNF);

((a &or b)&and(a&or&not(b)))&and(b&or&not(a))

Отже $a \wedge b = a \vee b \wedge a \vee \bar{b} \wedge b \vee \bar{a}$.

Команда **Contradiction(b)** перевіряє чи даний логічний вираз b є тотожно хибним.

> Contradiction(a &or &not(a &and b) &nor b);

true

Команда **Dual(b)** повертає логічний вираз, який дуальний до даного логічного виразу **b**, тобто вираз який утворений із **b** заміною **&and** на **&or** та **&or** на **&and**.

> Dual(a &implies b);

&not(a) &implies &not(b)

> Dual((&not a) &nor b &iff c);

(&not(a) &nand b) &xor c

Команда **Environment(level)** задає поточне оточення для автоматичного спрощення логічних виразів, тобто які тотожності і властивості будуть автоматично застосовуватися до логічного виразу під час дії команди. За умовчанням **level** рівний нулю. Рівні логічного спрощення можуть бути такими

1) **0** – немає спрощення.

2) **1** – опускаються надлишкові дужки; логічний вираз має бути виражений в термінах операцій **&and, &or** та **&not**.

3) **2** – додатково до рівня спрощення 1, застосовуються властивості (**a &and a --> a**) та (**a &or a --> a**) та знання логічних констант **true** і **false**.

Розглянемо на прикладі як буде спрощуватися логічний вираз $a \wedge b \wedge a$ при різних параметрах оточення

> Environment(0);

a &and b &and a;

(a &and b) &and a

> Environment(1);

a &and b &and a;

(a &and a) &and b

> Environment(2);

> a &and b &and a;

a &and b

Команда **Equivalent(a,b)** перевіряє, чи логічні вирази **a** та **b** є логічно еквівалентними. Якщо це так, то повертається значення **true** і у протилежному випадку повертається значення **false**.

> Equivalent(a &iff (a &or b), b &implies a);

true

> Equivalent(a &implies b, b &implies a, 'p');

false

Команда **Implies(a,b)** перевіряє чи істинний наступний логічний вираз $a \to b$. Якщо це так, то повертається значення **true** і у протилежному випадку повертається значення **false**.

> Implies(a &and b, a &implies b);

true

> Implies(&xor(a,b),&and(a,b)&or(&nota)&or(&notb));

true

Команда **Normalize(b,form)** перетворює логічний вираз **b** до певної нормальної форми. Тип нормальної форми визначається опцією **form**. Значеннями **form** може бути **DNF** для диз'юнктивної нормальної форми або **CNF** для кон'юнктивної нормальної форми. Якщо не задано жодної опції, то за умовчанням логічний вираз перетворюється до диз'юнктивної нормальної форми.

Перетворення виконуються із застосуванням розподільчого закону та законів де Моргана.

> Normalize(a &and (b &or c));

(a &and b) &or (a &and c)

> Normalize(&not(a &or b), form=CNF);

&not(a) &and &not(b)

Команда **Random(alpha, opt)** повертає випадковий логічний вираз від заданої множини букв **alpha** за-даний у певній канонічній формі **opt**. За умовчанням використовується диз'юнктивна нормальна форма.

> Random({a,b}, form=DNF);

((a&and&not(b))&or(b&and&not(a)))&or(&not(a)&and&not(b))

> Random({a,b}, form=CNF);

b &or &not(a)

Команда **Satisfy(expr)** повертає набір значень змінних при яких даний логічний вираз **expr** істинний. Якщо таких значень не існує то повертається значення **NULL**.

> Satisfy(a &or b);

{a=true, b=false}

Команда **Tautology(b)** перевіряє чи даний логічний вираз **b** є тотожньо істинним.

> Tautology(&and(a,b) &or (&not a) &or (&not b));

true

> Tautology((a &iff b) &or b);

false

Команда **TruthTable(b,set)** будує таблицю істинності для логічного виразу **b** і набору змінних **set**.

> T1 := TruthTable(a &xor b,[a,b]);

T1:=table([(true,true)=false, (true,false)=true,(false, true)=true,(false,false)=false])

Для того, щоб взнати чи вираз **b** приймає істинне чи хибне значення при певному наборі змінних із списку **set** потрібно викликати значення таблиці при цьому наборі.

> T1[true, false];

true

3. Висновки

В статті дано опис опис команд пакету logic системи комп'ютерної алгебри Maple. Розглянуто способи розв'язання деяких типових задач математичної логіки в Maple.

Використовуючи розглянуті команди пакету Maple можна ілюструвати розв'язання задач із курсу математичної логіки.

Література

1. Черняк А.А. Синтез классической и компьютерной математики в обучении /А.А.Черняк А.А., Ю.А.Доманова,Т.Н. Ранько//Информатизация образования. –№ 1. –2005. – С. 36-45.
2. Samková L, Calculus of one and more variables with Maple/L. Samková // International Journal of Mathematical Education in Science and Technology. – V. 43. –№2. –2012. –P.230-244.
3. Adym, E. (2005). The use of computers in mathematics education: A paradigm shift from "computer assisted instruction" towards "student rogramming"/E.Adym // The Turkish Online Journal of Educational Technology.- 4(2). –2005. – P.27–34.
4. Дьяконов В.П. Maple 9.5/10 в математике, физике и образовании/ В.П.Дьяконов — М.: С.Пресс, 2000. — 453 с.
5. Васильев А. Н. Maple 8. Самоучитель/Васильев А. Н.-М.: Диалектика, 2003.-352 с.
6. Кужель О . В. Елементи теорії множин і математичної логіки/О.В. Кужель– К:Рад. школа, 1977. – 160 с.
5. Лиман Ф. М. Математична логіка і теорія алгоритмів/Ф.М. Лиман – Суми: Вид-во „Слобожанщина", 1998. – 152 с.