КОМПЬЮТЕРНАЯ МАТЕМАТИКА

УДК 510.2

Д.О. ТЕРЛЕЦЬКИЙ

МНОЖИНИ ЯК СУКУПНОСТІ СУТНОСТЕЙ-ОБ'ЄКТІВ

Пропонується деяке розширення та формалізація класичної теорії множин у вигляді конструктивної її версії (CST) в якій вводиться новий рівень – рівень об'єктів, які є складовими для створення множин; концепція класу об'єктів, що дозволяє у певному сенсі формалізувати класифікацію самих об'єктів. У основу CST покладається принцип нескінченності за Брауером та відкидаються такі поняття «одноелементна множина» та «пуста множина». Проводиться порівняльний аналіз CST з деякими найбільш відомими системами теорії множин.

Вступ. Поняття «множина» є центральним для теорії множин і одним із фундаментальних понять для математики в цілому, але окрім цього це поняття має важливе та глобальне значення в життєдіяльності людини. Ми використовуємо множини повсякденно в своїй розумовій діяльності у процесі сприйняття, аналізу, порівняння, пошуку, класифікації тощо. Свідомо або підсвідомо створюємо множини, оперуємо ними, застосовуємо до них різні операції, зокрема, теоретико-множинні.

Огляд відомих систем теорії множин. З появою перших робіт, присвячених теорії множин [1], розпочалися активні дослідження в даному напрямку. Досить швидко цю теорію стали відносити до так званих основ математики, оскільки основне поняття цієї теорії – «множина» є одним із фундаментальних понять для математики. Протягом всієї історії розвитку даної теорії запропоновано багато варіантів теорії множин, це в першу чергу пов'язано з парадоксами, які були знайдені в канторівській теорії множин. Після чого відбулися деякі спроби побудувати теорію множин використовуючи підходи відмінні від запропонованого Кантором, так з'явилася теорія типів Рассела (T) [2], яку пізніше розвинув Куайн (NF), (ML) [3, 4]. Згодом відбулися спроби аксіоматизації теорії множин, які відомі в історії, як аксіоматичні теорії множин Цермело – Френкеля (ZF) та фон Неймана – Бернайса – Гєделя (NBG) [3 – 5].

В даний момент більшість математиків беруть за основу аксіоматичну систему теорії множин, яку запропонували Цермело та Френкель, проте відносно недавно з'явилася нова альтернативна теорія множин, яку запропонував Вопенка (AST) [6]. Ця теорія найбільш конструктивна серед вищезгаданих і це дуже важливий момент, оскільки для потреб математики, як показує практика, достатньо ZF, проте для комп'ютерних наук та штучного інтелекту просто теорії недостатньо, необхідно щоб вона була конструктивною. Саме конструктивність теорії є гарантією існування алгоритмів побудови різного роду об'єктів у її рамках.

Постановка проблеми. Аналізуючи вищезгадані варіації теорії множин, виникають питання про походження так званих «конкретних» множин. Виходячи з основ класичної теорії множин [7], можна зробити висновок, що «нові» множини можна отримувати шляхом теоретико-множинних операцій над «вже існуючими» множинами і це дійсно так. Проте не зникають питання про походження цих так званих «вже існуючих» множин, їхню кількість, їх види і т. д. Виникають питання про можливості автоматизації процесу створення множин для машини, без сторонньої допомоги; автоматизації класифікації та ідентифікації елементів множин; автоматичної генерації множин, що належать до певного класу. Тобто виникає питання про можливість практичної реалізації для машин здатності оперувати такою базовою категорією людського мислення як «множина».

Об'єкти та класи об'єктів. Відомо, що будь-яка множина складається з елементів, які її утворюють (визначають, формують). Елементами множини можуть виступати будь-які предмети, явища нашої уяви або навколишнього світу, для зручності називатимемо такі елементи об'єктами. Розглянемо такий об'єкт, як натуральне число, очевидно, що кожне натуральне число має бути цілим та додатнім, що є по суті характеристичними властивостями натуральних чисел, звідси не важко переконатися, що 1 – це дійсно натуральне число, а 5 або 2.5 – не є такими. Аналізуючи цей факт, можна прийти до висновку, що будь-який об'єкт володіє певними властивостями, які є для нього характери-стичними і визначають його, як деяку сутність та дозволяють відрізняти його серед інших об'єктів. Спираючись на це та все вищесказане сформулюємо визначення поняття «об'єкт».

Об'єкт A – це носій деякої сукупності (множини) властивостей та ознак $P(A) = (p_1(A),..., p_n(A))$, де $P(A)$ – вектор характеристичних властивостей об'єкта A, що визначають його, як деяку сутність. Позначатимемо довільний об'єкт наступним чином:

$$A = A / P(A) = A /(p_1(A),..., p_n(A)).$$

Для будь-яких об'єктів виконуються наступні аксіоми:

1) об'єкти існують незалежно від множин до яких вони входять або можуть входити;

2) кожен об'єкт може одночасно належати довільній кількості множин;

3) кожен об'єкт має певний не пустий набір властивостей, що визначає його, як деяку сутність.

На перший погляд дане визначення може здатися суперечливим, оскільки вище зазначено, що множину утворюють її елементи – об'єкти, але при визначенні об'єкта використовується поняття множини, яке ми ще не визначили. Виникає питання про різницю між поняттям «множина» та «множина властивостей деякого об'єкта». Довільний об'єкт A створюється за певною специфікацією, яка складається з властивостей $P(A) = (p_1(A),..., p_n(A))$, причому конкретні властивості $p_i(A)$ не можуть існувати окремо від об'єкта A, оскільки вони є його проявом та наочною демонстрацією, і спостерігати їх можна лише створивши самий об'єкт A – носій цих властивостей. У свою чергу об'єкт A не може існувати окремо від властивостей $P(A) = (p_1(A),..., p_n(A))$, оскільки без них невідомо, як його створювати. Отже властивість і об'єкт – нерозривно пов'язані між собою і не можуть існувати окремо один від одного. Відповідно, якщо припустити, що множина властивостей об'єкта A є множиною, то отримаємо суперечність, оскільки не відомо, як створювати конкретну властивість $p_i(A)$, де $i = \overline{1,n}$, якщо допускати, що $P(A)$ – це множина, то відповідно властивість $p_i(A)$ – має бути об'єктом цієї множини і повинна мати свій характеристичний вектор властивостей $p_i(A) = (p_{i_1}(A),..., p_{i_m}(A))$, що є неможливим, оскільки не зрозуміло які властивості мають бути у самої властивості об'єкта. Якщо розглянути властивість натурального числа «бути парним» або просто «парність», то виникають питання про властивості якими має володіти «парність», її створення з урахуванням того, що сама «парність» у даному випадку є об'єктом. Отже множина об'єктів і множина властивостей об'єкта – це нетотожні поняття.

Введемо такі поняття як «специфікація об'єкта», «розмірність об'єкта», «примітивний об'єкт» та «однотипні об'єкти».

Специфікація об'єкта A – це набір його характеристичних властивостей $P(A) = (p_1(A),..., p_n(A))$.

Розмірність об'єкта $D(A)$ – це деяке натуральне число n, яке визначає кількість властивостей об'єкта A, що входять до його специфікації $P(A) = (p_1(A),..., p_n(A))$.

Об'єкт A – примітивний тоді й тільки тоді, коли його розмірність $D(A) = 1$.

Об'єкти A і B – однотипні (різнотипні), тоді й тільки тоді, коли вони мають однакову (різну) розмірність і їм властива одна й та ж (не одна й та ж) специфікація.

Кожен об'єкт визначається певною специфікацією, але існують специфікації, що визначають декілька об'єктів одночасно, при цьому ці об'єкти можуть бути відмінними між собою, оскільки міра відповідності кожного конкретного об'єкта до специфікації може бути різною.

У загальному об'єкти можна поділити на конкретні (реально матеріально існуючі) та абстрактні. Кожен конкретний об'єкт, незалежно від того коли й як він був створений, є нічим іншим, як матеріальною реалізацією свого абстрактного образу – прототипу. Кожен прототип по суті, є абстрактною специфікацією для створення майбутніх реальних об'єктів. Окрім властивостей об'єктів також варто виділити операції (методи), які можна застосувати до об'єктів враховуючи особливості їх специфікації. У програмуванні [8, 9] часто використовують специфікацію та сигнатуру об'єкта без самого об'єкта називаючи це типом або класом об'єкта, тому використавши цю ідею сформулюємо визначання «клас об'єктів».

Клас об'єктів – це абстрактна специфікація та набір операцій (сигнатура) певної кількості певних об'єктів. Позначатимемо клас об'єктів наступним чином:

$$T = (P(T), F(T)) = ((p_1(T),..., p_n(T)), (f_1(T),..., f_k(T))),$$

де T – ім'я класу, $P(T)$ – абстрактна специфікація, а $F(T)$ – набір операцій (методів), які можна застосовувати до об'єктів класу T, враховуючи особливості їх специфікації.

Клас об'єктів є однорідним (неоднорідним) тоді й тільки тоді, коли за його специфікацією можна побудувати однотипні (різнотипні) об'єкти.

Важливо розуміти, що клас T – це специфікація $P(T) = (p_1(T),..., p_n(T))$ та сигнатура $F(T) = (f_1(T),..., f_k(T))$ деякого абстрактного об'єкта A_T, тобто клас T по суті – це деякий абстрактний об'єкт, A_T – прототип, на основі якого можна створювати конкретні об'єкти. Коли говоримо про деякий клас об'єктів, то маємо на увазі властивості цих об'єктів та методи які можна до них застосувати, тобто клас – це ніщо інше, як узагальнена форма розгляду властивостей об'єктів та операцій над об'єктами без самих об'єктів.

Використовуючи поняття однорідного класу, можна визначити примітивні типи даних для деяких мов програмування. Для прикладу, визначимо тип Int у мові програмування C++. Отже, визначатимемо клас Int, який має специфікацію $P(Int) = (p_1(Int),\ p_2(Int))$, де властивість $p_1(Int)$ – це «цілі числа», а властивість $p_2(Int)$ – це «числа не більші, ніж 2147336147 та не менші, ніж − 2147336148».

Очевидно, що всі числа, які володітимуть властивостями $p_1(Int)$ та $p_2(Int)$ будуть об'єктами класу Int. Тепер можна визначити сигнатуру класу Int, вона може містити довільну кількість різних методів, для прикладу визначимо елементарні операції над цілими числами:

$$F(Int) = (f_1(Int), f_2(Int), f_3(Int), f_4(Int)),$$

де $f_1(Int) = "+"$, $f_1(Int) = "-"$, $f_1(Int) = "\cdot"$, $f_1(Int) = "\div"$.

У випадку неоднорідного класу ми маємо ситуацію коли в рамках одного класу можуть існувати об'єкти різної розмірності з довільними специфікаціями. Для прикладу розглянемо клас дійсних чисел R, він володіє специфікацією $P(R) = (p_1(R), p_2(R), p_3(R), p_4(R), p_5(R),...)$, де наприклад, $p_1(R)$ = = «цілі числа», $p_2(R)$ = «натуральні числа», $p_3(R)$= «дробові числа», $p_4(R)$ = «від'ємні цілі числа», $p_5(R)$ = «парні числа». Зрозуміло, що клас R має дуже багато властивостей і що його специфікація $P(R)$ досить велика, але щоб показати специфіку неоднорідних класів ми розглянемо лише деякі, вибрані властивості з цієї специфікації.

Розглянемо числа 1, 2.75, −16, 4, −7.48. Всі вони є об'єктами класу R, але якщо порівняти їх між собою, то очевидно, що це різнотипні об'єкти. Але оскільки ці об'єкти належать до класу R, то вони мають відповідати специфікації класу $P(R) = (p_1(R), p_2(R), p_3(R), p_4(R), p_5(R),...)$. Зрозуміло, що кожне з чисел, які ми розглядаємо, відповідатиме специфікації класу по своєму. Табл. 1 показує відповідність специфікацій об'єктів до специфікації класу R. Зрозуміло, що всі ці числа 1, 2.75, −16, 4, −7.48 – об'єкти класу, оскільки специфікація жодного з них не суперечить специфікації класу R, більше того, специфікація класу R включає у себе специфікації всіх об'єктів класу, тобто: $P(1) \in P(R)$, $P(2.75) \in P(R)$, $P(-16) \in P(R)$, $P(4) \in P(R)$, $P(-7.48) \in P(R)$.

ТАБЛИЦЯ 1. Специфікація об'єктів 1, 2.75, −16, 4, −7.48 у класі R

$A_i \,/\, p_j(A_i)$	Ціле число	Натуральне число	Дробове число	Від'ємне число	Парне число	…
1	1	1	0	0	0	
2.75	0	0	1	0	0	
–16	1	0	0	1	1	
4	1	1	0	0	1	
–7.48	0	0	1	1	1	

Зрозуміло, що існує не один варіант визначення специфікації класу R, різниця між ними полягає лише в деталізації самої специфікації класу. Підтвердженням цього є те, що властивість $p_2(R)$ = «натуральні числа» можна було б представити, як $p_{21}(R) \cup p_{22}(R)$, де $p_{21}(R)$ = «цілі числа», а $p_{22}(R)$ = «додатні числа», оскільки властивості $p_{21}(R)$, $p_{22}(R)$ утворюють деталізовану специфікацію натурального числа. Аналогічно можна було б деталізувати властивість $p_1(R)$ = «цілі числа» представивши її як $p_{11}(R) \cup p_{12}(R) \cup p_{13}(R)$, де $p_{11}(R)$ = «числа, що не мають дробової частини», $p_{12}(R)$ = «додатні числа», а $p_{13}(R)$ = «від'ємні числа».

Клас об'єктів T_1 – це підклас класу T тоді й тільки тоді, коли $D(T_1) \leq D(T)$ і $P(T_1) \subseteq P(T)$.

Розглянемо наступні класи: клас цілих чисел Z, клас натуральних чисел N та клас парних натуральних чисел N_2. Опишемо їх наступними специфікаціями:

$P(Z) = (p_1(Z), p_2(Z))$, $P(N) = (p_1(N), p_2(N))$, $P(N_2) = (p_1(N_2), p_2(N_2))$, де $p_1(Z)$ = «цілі числа», $p_2(Z)$ = «додатні або від'ємні числа»; $p_1(N)$ = «цілі числа», $p_2(N)$ = «додатні числа»; $p_1(N_2)$ = «цілі числа», $p_2(N_2)$ = «додатні парні числа». Якщо порівняти між собою класи N і N_2 то можна помітити, що $p_1(N) = p_1(N_2)$ і $p_2(N) \neq p_2(N_2)$, але ми знаємо, що додатні числа включать у себе додатні парні числа, тобто $p_2(N_2) \subset p_2(N)$, звідси випливає, що $N_2 \subset N$, тобто клас N_2 – однорідний підклас класу N. Тепер порівняємо класи N та Z, очевидно, що $p_1(N) = p_1(Z)$ і $p_2(N) \neq p_2(Z)$, але аналогічно, як і в попередньому випадку $p_2(N) \subset p_2(Z)$), відповідно $N \subset Z$, з чого випливає, що $N_2 \subset N \subset Z$.

Множини об'єктів. Спираючись на поняття «об'єкт» і «клас об'єкта», сформулюємо визначення поняття «множина об'єктів».

Множина об'єктів S – це об'єднання, що задовольняє одну із наступних схем: 1) $O_1 \cup ... \cup O_n = S$; 2) $S_1 \cup ... \cup S_m \cup O_1 \cup ... \cup O_n = S$; 3) $S_1 \cup ... \cup S_m = S$, де O_i $i = \overline{1,n}$ – це об'єкти, а S_j $j = \overline{1,m}$ – це множини об'єктів, при цьому O_i та S_j можуть відноситися до різних класів.

Очевидно, що в залежності від класу O_i та S_j залежить клас новоствореної множини об'єктів. Розглянемо деяку множину S_1, яка складається з 5-ти натуральних чисел $S_1 = \{1, 2, 3, 4, 5\}$. Кожне число є об'єктом, що володіє певною специфікацією, у даному випадку всі об'єкти володіють однією і тією ж специфікацією $P(S_1) = (p_1(S_1), p_2(S_1))$, де $p_1(S_1)$ = «ціле число», а $p_2(S_1)$ = «додатне число». Специфікація $P(S_1)$ утворює клас N_{S_1}, що є однорідним підкласом класу N – класу натуральних чисел, тобто в даному випадку множина S_1, що складається з однорідних об'єктів, є множиною класу N_{S_1}.

Розглянемо деяку множину S_2, яка складається з 5-ти дійсних чисел $S_2 = \{0.24, 4, -3, 0.31, -2.37\}$, відповідно всі ці числа є різнотипними об'єктами, що володіють різними специфікаціями. Очевидно, що множина S_2 – множина класу R_{S_2}. В табл. 2 приведено неоднорідність класу R_{S_2}, який є неоднорідним підкласом класу R – класу дійсних чисел.

ТАБЛИЦЯ 2. Специфікації об'єктів множини S_2

$A_i / p_j(A_i)$	Ціле число	Дотане число	Парне число	…
0.2	дробове	додатне	парне	

4	ціле	додатне	парне	
–3	ціле	від’ємне	непарне	
0.31	дробове	додатне	непарне	
–2.37	дробове	від’ємне	непарне	

У класичній теорії множин викладеній у [7] розглядається утворення множин лише за схемою 3) $S_1 \cup S_2 = S_3$, при цьому вважається що S_1 та S_2 уже створені. В рамках пропонованої автором конструктивної теорії множин (CST) існує ще дві схеми 1) $O \cup O = S$ та 2) $S \cup O = S$, які виступають певним поясненням утворення множин S_1 та S_2 для схеми 3). Дані схеми носять конструктивний характер і в певному сенсі абстрактною формою алгоритмів створення (генерації) множин.

Висновки. Одним із базових понять CST є поняття класу, яке також відіграє важливу роль в T, NF та ML. Проте на відміну від T, NF та ML, CST не має строгої ієрархії типів та допускає існування класів, які містять як однотипні (однорідні класи), так і різнотипні об’єкти (неоднорідні класи). В CST відсутнє поняття «елемент класу», натомість вводиться поняття «об’єкт класу» та розглядаються три типи належності об’єкта до класу. Також в CST клас не завжди може мати більшу розмірність ніж об’єкт, це випливає з визначень однорідного та неоднорідного класу.

У порівнянні з ZF в CST відсутнє поняття пустої та одноелементної множини, натомість вводиться поняття об’єкта, хоча в основі обох теорій лежить принцип нескінченності за Брауером. Важлива особливість CST – те, що вона як і ZF має механізм для побудови множин, проте на відміну від ZF, CST має три схеми побудови множин 1) $O \cup O = S$, 2) $S \cup O = S$, 3) $S \cup S = S$, де O – це об’єкт, а S – це множина об’єктів, при цьому O та S можуть відноситися до різних класів. Наявність таких схем дозволяє говорити не лише про конструктивність CST, а й про те що CST в деякому сенсі конструктивно пояснює існування так званих «уже існуючих» множин і дозволяє працювати як з вже заданими множинами, так і створювати або задавати нові.

На відміну від NBG, де всі об’єкти – це класи, в CST поняття об’єкта, класу об’єкта та множини об’єктів вводяться окремо і не є тотожними. В NBG клас відповідає нашому інтуітивному розумінню сукупності, а поняття «множина» належить лише до тих класів, які самі є елементами інших класів. В CST відсутнє поняття «елемент класу», проте вводиться поняття підкласу і розглядаються множини об’єктів певного класу. При цьому розрізняються поняття «об’єкт класу», «клас об’єкта», «множина класу» та «клас множини». В NBG також існують класи які не є множинами, їх називають власними класами, що дозволяє провести певну аналогію з CST, де елементи які не є множинами називаються об’єктами. Також важливою особливістю CST є те, що елементами множини можуть бути лише об’єкти, хоча при цьому вводиться поняття підмножини.

Одним з центральних понять AST і CST є поняття об’єкта, в чому власне проявляється схожість між цими теоріями, проте, як і в ZF, в AST вводиться поняття пустої множини, яке відсутнє в CST. Проте в CST, як і в AST множини конструюються з об’єктів. CST як і AST не допускає актуальної нескінченності. Також в AST вводиться поняття універсаму множин, що були сконструйовані ітеративно, починаючи з пустої множини, тобто в AST постулюється існування деяких вже наперед створених множин. В CST множини «з'являються» лише після конструктивної побудови, мається на увазі існування алгоритмів для побудови множин об'єктів. Серед усіх вищезгаданих аксіоматичних систем теорії множин AST є найбільш близькою до CST, хоч і має деякі принципові відмінності.

Підводячи підсумки, можна зробити висновок, що запропонована система CST увібрала в себе деякі ідеї з T, ZF, NBG та AST, але водночас вона має принципові відмінності від цих теорій. CST дуже перспективна і конструктивна теорія, яка дозволяє будувати та одночасно класифікувати множини різних класів, проте вона потребує подальшого, більш глибокого дослідження, яке буде проведене автором в наступних роботах.

Д.А. Терлецкий

МНОЖЕСТВА КАК СОВОКУПНОСТИ СУЩНОСТЕЙ-ОБЪЕКТОВ

Предлагается некоторое расширение и формализация классической теории множеств в виде конструктивной ее версии (CST), в которой вводится новый уровень – уровень объектов, являющихся составляющими для создания множеств; концепция класса объектов, позволяет в определенном смысле формализовать классификацию самих объектов. В основу CST возлагается принцип бесконечности по Брауэру и отвергаются понятия «одноэлементное множество» и «пустое множество». Проводится сравнительный анализ CST с некоторыми наиболее известными системами теории.

D.O. Terletskyi

CONSRUCTIVE SETS I

In this paper some expansion and formalization of classical set theory as it's constructive version (CST) are proposed. A new level – the level of objects that are components for sets creating, and concept of a class of objects, that can in some sense to formalize the classification of the objects themselves introduced within CST. (CST) relies in the basis on the principle of infinity by Brouwer. Such notions like «singleton set» and «empty set» rejected in CST. The comparative analysis of the CST with some of the most known systems of set theory also proposes by author in this article.

1. *Кантор Г.* Труды по теории множеств. – М.: Наука, 1985. – 430 с.
2. *Russell B.* Mathematical Logic as Based on the Theory of Types // American Journal of Mathematics. – 1908. –Vol. 30, N. 3. – P. 222–262.
3. *Френкель А.А., Бар-Хиллел И.* Основания теории множеств. – М.: Мир, 1966. – 556 с.
4. *Ван Хао, Р. Мак-Нортон.* Аксиоматические системы теории множеств. – М.: Изд-во иностран. лит., 1963. – 56 с.
5. *Голдблатт Р.* Топосы. Категорный анализ логики. – М.: Мир, 1983. – 488 с.
6. *Вопенка П.* Математика в альтернативной теории множеств. – М.: Мир, 1983. – 154 с.
7. *Столл Р.Р.* Множества. Логика. Аксиоматические теории. – М.: Просвещение, 1968. – 230 с.
8. *Эккель Б.* Философия Java. 4-е издание. – Питер, 2009. – 638 с.
9. *Страуструп Б.* Язык программирования С++. Спец. изд. Невский диалект, 2008. – 1054 с.

Про автора:

Терлецький Дмитро Олександрович,

аспірант факультету кібернетики
Київського національного університету імені Тараса Шевченка.
E-mail: dmytro.terletskyi@gmail.com