

Використання пакету Wolfram Demonstrations Project у навчальному процесі надає можливість переглядати у динаміці виконання алгоритмів розв'язування задач дискретної математики, що значно полегшує розуміння їх сутності, моделювання досліджуваних явищ, допомагає процесу висунення гіпотез; надає можливість здійснювати дослідження та перевірку результатів; сприяє посиленню інтересу студентів до вивчення дисципліни та формуванню фахових компетентностей.

Література

1. Жалдак М. І., Рамський Ю. С., Рафальська М. В. Модель системи соціально-професійних компетентностей вчителя інформатики. // Науковий часопис Національного педагогічного університету імені М. П. Драгоманова. Серія 2: комп'ютерно-орієнтовані системи навчання. – К.: НПУ імені М. П. Драгоманова., 2009. – №7(14) – С. 3-10.
2. Жалдак М. І. Система підготовки вчителя до використання інформаційно-комунікаційних технологій в навчальному процесі // Науковий часопис Національного педагогічного університету імені М.П. Драгоманова. Серія 2: комп'ютерно-орієнтовані системи навчання: Зб. наук. праць – К.: НПУ імені М.П. Драгоманова., 2011. – №11(18) – С. 3-16.
3. Wolfram Research [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Wolfram_Research.
4. Wolfram Demonstrations Project [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Wolfram_Demonstrations_Project.
5. Wolfram Demonstrations Project [Електронний ресурс]. – Режим доступу: <http://demonstrations.wolfram.com>.

Костюченко А.О.

Чернігівський національний педагогічний університет імені Т.Г.Шевченка

Особливості створення педагогічних програмних засобів математичного спрямування

В процесі створення педагогічного програмного засобу (ППЗ), який буде використовуватися в межах деякої предметної галузі, розробник має розумітися на термінах, поняттях, правилах, методах опрацювання даних, що стосуються цієї предметної галузі.

Можна визначити кілька особливостей, які необхідно враховувати під час створення математичних ППЗ а саме:

- наявність відповідних математичних знань у розробника;
- можливість обчислення за допомогою ППЗ значень заданих користувачем математичних та логічних виразів;
- можливість проведення символічних обчислень за допомогою ППЗ;
- врахування похибок обчислень;
- виведення математичних виразів в “природному” поданні;
- виконання графічних побудов.

Необхідною умовою створення математичних ППЗ є наявність у студента чи викладача, який буде створювати таке ППЗ, необхідної бази математичних знань з відповідної предметної галузі.

Під час використання математичних ППЗ у більшості випадків користувачеві необхідно працювати з математичними та логічними виразами, заданими власноруч. В зв'язку з чим постають задачі опрацювання цих виразів. Такими задачами можуть бути:

- обчислення значень математичних виразів з конкретними числовими значеннями (наприклад, $2+5*\sin(30)$);
- обчислення значень функцій від однієї змінної (наприклад, $\sin(x)+\cos(x)$, $x*\sin(x)+5$) для заданого значення аргумента;
- визначення істинності чи хибності логічних виразів (наприклад, $\sin(3+5)>0$, $(\sin(x)<\cos(x))$ or $(\cos(x)>0)$) і для заданих значень змінних.

Для опрацювання такого виразу необхідно здійснити синтаксичний аналіз текстового рядка, де міститься вхідний вираз, і перетворення текстових даних в спеціальну форму запису. Таке перетворення можна реалізувати кількома методами, серед яких: подання виразу у формі Бекуса-Наура, метод рекурсивного спуску та подання виразу у формі польського запису (у польській нотації) чи польського інверсного (зворотного) запису.

Більш детально зупинимося на використанні форми польського запису та форми польського інверсного запису, які є легшими для розуміння та реалізації. Форма запису має назву «польський запис» на честь польського логіка Яна Лукасевича, яку він винайшов в 1920 році, щоб спростити логіку висловлювань. Польська інверсна форма запису була розроблена Чарльзом Хембліном в

середині 1950-х років на основі польської нотації. Особливістю даної форми запису є те що, символи операцій завжди йдуть відокремлено від своїх операндів (до або після), та відсутність дужок, а порядок виконання операцій визначається в залежності від розміщення операторів і є однозначним. Причому обчислення виразів, записаних в такій нотації, можна проводити шляхом одноразового перегляду з використанням стеку.

«Традиційна» форма запису математичних виразів, називається інфіксною формою запису, тобто являє собою запис математичного виразу, в якому знак операції знаходиться між його операндами (наприклад, $(5 - 6) * 7$).

Польський запис називається префіксною формою запису, тобто запис математичного виразу, в якому знак операції розміщений перед його операндами; (наприклад, $* - 5 6 7$).

Польський інверсний запис називається постфіксною формою запису, тобто запис математичного виразу, в якому операнди розміщені перед знаком операції; (наприклад, $5 6 - 7 *$).

Під час формування польського чи інверсного польського запису, в яких дужки не використовуються, важливо враховувати пріоритети арифметичних операцій:

- знак піднесення до степеня “^” – 4-й пріоритет;
- знаки множення та ділення “*”, “/” – 3-й пріоритет;
- знаки додавання та віднімання “+”, “-” – 2-й пріоритет.

1-й пріоритет досить часто віддається дужкам “(” та “)”, а 0-й пріоритет відносять до операндів.

У випадку, якщо у виразі містяться математичні функції (наприклад: ln, log, sin, cos, abs, ...), під час побудови польського запису їх необхідно розглядати як оператори з максимальним пріоритетом. А в процесі обчислення значення виразу за даним польським записом функції необхідно розглядати як унарні операції з максимальним пріоритетом. Наприклад, вираз « $\sin(3+5*x)+\cos(x)+\log(\ln(x+1))$ » буде записаний в інверсній польській формі, як « $3 5 x * + \sin x \cos x 1 + \ln \log * +$ ».

Для прикладу розглянемо алгоритми опрацювання польського інверсного запису, які може використовувати розробник.

Побудова польського інверсного запису за інфіксною формою запису:

- Переглядаємо вхідний рядок зліва вправо.
 - Якщо зустрічається операнд, то копіюємо його у вихідний рядок
 - Якщо зустрічається відкриваюча дужка, то розміщуємо її в стек
 - Якщо зустрічаємо знак операції, то
 - Якщо стек порожній, то розміщуємо знак операції в стек
 - Інакше
 - Якщо пріоритет останньої операції в стеці нижче пріоритету поточної операції, то розміщуємо знак операції в стек
 - Інакше
 - Записуємо знаки операції із стеку в вихідний рядок до тих пір, поки пріоритет останньої операції в стеці буде не нижчий пріоритету поточної операції
 - Розміщуємо знак поточної операції в стек
 - Якщо зустрічається закриваюча дужка, то до тих пір поки верхнім елементом не стане відкриваюча дужка, виштовхуємо елементи з стеку у вихідний рядок. В такому разі відкриваюча дужка вилучається з стеку, але у вихідний рядок не додається дужка, що закривається
- Записуємо знаки операції з стеку в постфіксний запис до тих пір, поки стек не порожній.

Розглянемо застосування зазначеного алгоритму на конкретному прикладі. Нехай дано математичний вираз в інфіксній формі « $a * (b + c) - d$ », який потрібно перевести в постфіксну форму. Покрокове виконання переведення подано в таблиці 1.

Таблиця 1.

Переведення виразу з інфіксної в постфіксну форму

Елемент для аналізу	Виконувана операція	Результуючий рядок	Елементи стеку (вершина праворуч)		
a	Додаємо «a» до результуючого рядка	a			
*	Розміщуємо «*» в стек	a	*		
(Розміщуємо «(» в стек	a	*	(

b	Додаємо «b» до результуючого рядка	a b	*	(
+	Розміщуємо «+» в стек	a b	*	(+
c	Додаємо «c» до результуючого рядка	a b c	*	(+
)	Виштовхуємо з стеку оператори до результуючого рядка, поки дійдемо до відкриваючої дужки. Дужку вилучаємо з стеку	a b c +	*		
-	Оскільки пріоритет останньої операції в стеці «*» вищий за пріоритет операції «-», то виштовхуємо операцію «*» з стеку в результуючий рядок. Розміщуємо «-» в стек	a b c + *	-		
d	Додаємо «d» до результуючого рядка	a b c + * d	-		
	Оскільки вхідний рядок порожній, виштовхуємо операції з стеку в результуючий рядок	a b c + * d -			

В результаті отримуємо запис в постфіксній формі заданого виразу: «a b c + * d -».

Обчислення значення виразу, поданого в постфіксній формі.

- Переглядаємо постфіксний запис зліва вправо
 - Якщо зустрічається операнд, то розміщуємо його в стек
 - Якщо зустрічається знак операції, то
 - Виконуємо цю операцію, використовуючи в якості операндів два числа з вершини стека
 - Результат розміщуємо в стеці
- В вершині стеку знаходиться результат обчислення

Розглянемо застосування зазначеного алгоритму на конкретному прикладі. Обчислити значення виразу «2 3 5 + * 1 -», заданого в постфіксній формі. Покрокове виконання обчислення подано в таблиці 2

Таблиця 2.

Обчислення значення виразу, заданого в постфіксній формі

Елемент для аналізу	Виконувана операція	Елементи стеку (вершина праворуч)		
2	Розміщуємо «2» в стек	2		
3	Розміщуємо «3» в стек	2	3	
5	Розміщуємо «5» в стек	2	3	5
+	Виконуємо операцію «+» над двома числами з вершини стеку «3» та «5», результат «8» розміщуємо в стек	2	8	
*	Виконуємо операцію «*» над двома числами з вершини стеку «2» та «8», результат «16» розміщуємо в стек	16		
1	Розміщуємо «1» в стек	16	1	
-	Виконуємо операцію «-» над двома числами з вершини стеку «16» та «1», результат «15» розміщуємо в стек	15		
	Оскільки вхідний рядок порожній, результат обчислення в стеці	15		

В результаті отримуємо, як результат, число 15.

Перетворення запису виразу, поданого в постфіксній формі, у інфіксну форму.

- Переглядаємо постфіксний запис зліва вправо
 - Якщо прочитано операнд, то заносимо його в стек
 - Якщо прочитано знак операції, то
 - Беремо два верхніх елементи з стеку
 - Якщо в першому елементі пріоритет операції менший (і не рівний нулю), ніж пріоритет розглядуваної операції, то беремо перший елемент в дужки
 - Якщо в другому елементі пріоритет операції менший (і не рівний нулю), ніж пріоритет розглядуваної операції, то беремо другий елемент в дужки

◦ Записуємо в стек рядок виду: 2-й елемент+знак операції+1-й елемент

• В вершині стеку знаходиться результат перетворення.

Розглянемо застосування зазначеного алгоритму на конкретному прикладі. Нехай дано запис математичного виразу в постфікській формі «a b c + * d -», який потрібно перевести в інфіксну форму. Покрокове виконання переведення подано в таблиці 3.

Таблиця 3.

Переведення виразу з постфіксної в інфіксну форму

Елемент для аналізу	Виконувана операція	Елементи стеку (вершина праворуч)		
a	Розміщуємо «a» в стек	a		
b	Розміщуємо «b» в стек	a	b	
c	Розміщуємо «c» в стек	a	b	c
+	Беремо 2 верхні елементи стеку «c» та «b»; оскільки вони є звичайними операндами (пріоритет рівний нулю), то в стек вносимо рядок «b+c»	a	b+c	
*	Беремо 2 верхні елементи стеку «b+c» та «a». «b+c» елемент з операцією «+», отже він має пріоритет операції «+», який нижчий за пріоритет операції «*», тому беремо його в дужки. «a» звичайний операнд (пріоритет рівний нулю). в стек вносимо рядок «a *(b+c)»	a*(b+c)		
d	«d» в стек	a*(b+c)	d	
-	Беремо 2 верхні елементи стеку «d» та «a*(b+c)». «d» звичайний операнд (пріоритет рівний нулю). «a*(b+c)» елемент з операцією «*» який вищий за пріоритет операції «-». в стек вносимо рядок «a*(b+c)-d»	a*(b+c)-d		
	Оскільки вхідний рядок порожній, то результат обчислення в стеці	a*(b+c)-d		

В результаті отримаємо запис в інфікській формі заданого виразу: «a * (b + c) – d».

Розглянуті вище алгоритми будуть стосуватися і логічних виразів. Пріоритети логічних операторів визначаються наступним чином:

- операції порівняння: (> ≥ = ≠ ≤ <) – 5-й пріоритет;
- логічне “ні” (not) – 4-й пріоритет;
- логічне “і” (and) – 3-й пріоритет;
- логічне “або” (or) – 2-й пріоритет.

В деяких випадках користувачеві, під час використання ППЗ потрібно провести символічні перетворення виразів, реалізація алгоритмів яких є досить складною задачею, наприклад відшукання похідної чи первісної (невизначеного інтеграла) для заданої функції, а інколи і нерозв’язною. Такі перетворення виразів також можуть базуватися на використанні польських форм записів.

На даний час існує значна кількість вже готових програмних засобів для виконання обчислень математичних та логічних виразів і проведення символічних обчислень, що можуть використовуватися розробниками під час створення власних програмних засобів. Проте створення власних реалізацій зазначених обчислень є досить ґрунтовною практикою для розробника і в подальшому може надати можливість розширення та вдосконалення розроблених алгоритмів.

В процесі виконання математичних обчислень виникають похибки таких обчислень. В процесі реалізації певного алгоритму виникає похибка, яка включає в себе: похибки виконання операцій; похибку методу, яка впливає з дискретного характеру будь-якого чисельного алгоритму; похибки округлення результатів, які пов’язані з використанням в обчислювальних машинах чисел з обмеженою точністю подання. Якщо ж розглядати застосування ППЗ для аналізу певної моделі, то джерелами похибок будуть ще похибки математичної моделі, пов’язані з її неповною відповідністю реаліям предметній галузі, та похибки вхідних даних, прийнятих для розрахунку.

В процесі використання ППЗ математичного призначення може виникнути необхідність у відображенні математичних виразів у “природному” вигляді. В залежності від формату документу, в який буде проводитися виведення, розв’язання даної задачі може бути різним. Так виведення може проводитися:

- в HTML-документ, в який буде генеруватися вигляд необхідного математичного виразу, використовуючи: таблиці для задання різних рівнів формули; надрядкові та підрядкові символи, для вказування степенів та індексів; зміну шрифтів для відображення грецьких та спеціальних символів;
- в документ з TEX форматуванням чи форматування окремих формул, використовуючи TEX розмітку для їх розміщення в HTML документі;
- в документ текстового процесора MS Word засобами microsoft equation чи документ офісного пакету LibreOffice, використовуючи його за рахунок COM технології.
- в документ графічного типу із збереженням в файлі і можливістю його подальшого використання.

Для виконання такого відображення математичних формул розроблена значна кількість бібліотек та розширень, наприклад: alTex, Ritex, ExprDraw, ExprMake.

Під час розробки ППЗ, в якому буде необхідно проводити графічні побудови (побудова графіків функцій, побудова геометричних об’єктів), необхідно пам’ятати про деякі особливості використання комп’ютера для виконання графічних побудов, а саме: відмінність системи координат графічного поля виведення від звичної декартової системи координат та необхідність виведення результатів графічних побудов на пристрій з дискретною кількістю точок.

Під час використання графічних побудов з використанням комп’ютера доводиться працювати з двома системами координат. Перша система – це система координат графічного поля виведення чи екранна система координат. Координатами точки в цій системі є номер пікселя в рядку X та номер рядка пікселів Y , де $0 \leq X \leq X_{max}$, $0 \leq Y \leq Y_{max}$. Початок координат розміщується в лівому верхньому кутку графічного поля виведення. Параметри екранної системи координат (максимальне число пікселів в рядках X_{max} та максимальне число рядків пікселів Y_{max}) залежать від поточного графічного режиму та розміру графічного поля виведення. Таким чином, ця система координат певним чином пов’язана з конкретним графічним пристроєм та режимом його роботи.

Друга система координат – так звана декартова або математична система координат (x, y) , де $x_{min} \leq x \leq x_{max}$ та $y_{min} \leq y \leq y_{max}$, яку визначає програміст і яка є незалежною від конкретного графічного пристрою виведення.

Параметри, через які задають діапазони зміни x і y (x_{min} , y_{min} , x_{max} , y_{max}), визначається прямокутна область в математичному двохвимірному просторі. Ці параметри залежать лише від конкретної задачі.

Математичні координати і координати графічного поля виведення зв’язані між собою простим відношенням:

$$X = X_{max} \cdot \left(\frac{x - x_{min}}{x_{max} - x_{min}} \right) \quad Y = Y_{max} \cdot \left[1 - \left(\frac{y - y_{min}}{y_{max} - y_{min}} \right) \right].$$

Формула для екранної координати Y дещо відрізняється від формули для координати X в силу того, що в екранній системі координат OY направлена вниз.

Що стосується другої особливості, то вона полягає в тому, що монітор, на якому буде відображатися побудова на деякому графічному полі – це прилад з дискретною кількістю точок. У нього скінченна ширина та висота і між будь-якими двома точками міститься скінченна і за математичними мірками, невелика кількість так званих елементів зображення – пікселів. Окрім того, координати всіх точок, які можна відобразити на екрані монітора, повинні бути цілочисельними. Природним виходом з даної ситуації є округлення отриманої координати точки до найближчого цілочисельного значення. Зрозуміло, що під час виконання такі дії відбувається часткове спотворення зображення графіка функції чи графічного об’єкта.

Література

1. Жалдак М.І., Горошко Ю.В., Вінниченко Є.Ф., Математика з комп’ютером. – Київ.: РННЦ ДНІТ, 2004. – 206 с.
2. Лук’янова В.В. Комп’ютерний аналіз даних: Посібник. – К.: Видавничий центр "Академія", 2003. – 344с.
3. Польський інверсний запис [Електронний ресурс] // Вікіпедія – вільна енциклопедія. – Режим доступу: http://uk.wikipedia.org/wiki/Польський_інверсний_запис (20.11.2013). – Назва з екрану.