

VISUALIZATION OF RECURSIVE FUNCTION BEHAVIOUR ⁸

Emilia Golemanova, PhD

Department of Computer Systems and Technologies,

“Angel Kanchev” University of Ruse

Tel.: 082-888-681

E-mail: EGolemanova@uni-ruse.bg

Tzanko Golemanov, PhD

Department of Computer Systems and Technologies,

“Angel Kanchev” University of Ruse

Tel.: 082-888-681

E-mail: TGolemanov@uni-ruse.bg

***Abstract:** Recursion is a concept in programming that has traditionally been difficult for students to understand, as well as challenging for instructors to teach. The reason is the complexity of the abstracted mental model of the recursive flow of control. This article presents a visually oriented approach to teaching recursion that uses visual notation for recursive function execution and specialized learning tool developed to help students understand the behavior of a recursive function. It interactively demonstrates three different types of recursive function: with a single recursive call (linear recursion), with two recursive calls in an expression, and with a recursive call in a loop (e.g. backtracking algorithm). Students can follow the execution flow step by step, visualizing the pseudo-code of the function, its forward and backward trace, and the recursive tree/control stack.*

***Keywords:** Recursion, Program Visualization, Recursive Function.*

ВЪВЕДЕНИЕ

Ускореното развитие на програмните технологии, в това число използването на Изкуствен интелект за генериране на програмен код, задълбочава противоречието между статичния запис на програмата, като последователност от инструкции, и динамичния характер на нейното изпълнение. Все по-съществен става проблемът с формирането у обучаемите на вътрешен визуален образ на изчислителния процес. Това се проявява особено остро при изучаването на такова сложно и абстрактно понятие като рекурсия.

Този доклад представя визуално ориентиран подход към преподаването на рекурсия, който използва визуална нотация за изпълнение на рекурсивна функция, и специализиран инструмент за обучение, разработен за да помогне на обучаемите да разберат поведението на рекурсивна функция.

ИЗЛОЖЕНИЕ

Рекурсия

Рекурсията е метод в програмирането за дефиниране на подпрограма или структура от данни чрез себе си. Така тя се превръща в техника за организиране на повторения:

- чрез рекурсивна подпрограма може да се опише безкраен брой повторения, дори ако тази подпрограма не съдържа явни цикли – рекурсивна функция;
- чрез рекурсивна структура от данни може да се дефинира безкрайно множество от обекти чрез краен оператор - списъци, дървета и графи.

Затова рекурсията е мощен механизъм. Същевременно с това тя е и красива, елегантна или както казва Лоурънс Питър Дойч – божествена, защото превръща програмирането в изкуство.

⁸ The paper was presented on 24 October 2025 in section “Communication and Computer Technologies” with original title in Bulgarian: ВИЗУАЛИЗАЦИЯ НА ПОВЕДЕНИЕТО НА РЕКУРСИВНА ФУНКЦИЯ

Рекурсията е важна идея в компютърните науки, но традиционно е била трудна за разбиране концепция за студентите, както като управляваща структура, така и като аналитичен инструмент, т.е. студентите имат трудности както с изпълнението, така и с разработката на рекурсивна функция. Основният проблем е с предаването на контрола между извикваща и извиквана функция. Рекурсивната функция извиква себе си, следователно обучаемите трябва да управляват различни екземпляри на една и съща функция (Tessler, 2013). Другият основен проблем е неразбирането на фазата на връщане (т. нар. обратен поток на управление, *passive flow*) в процеса на изпълнение на една рекурсивна функция. Те погрешно смятат, че след като е достигнат базовият случай, процесът спира. Накратко, те никога не си представят обратния поток на управление на изчислението. Другата съществена трудност, която ние наблюдаваме, е поведението във времето на входно-изходните параметри на функцията.

Рекурсията е предизвикателство и за преподаване, защото:

- не е понятие, на което лесно може да се намери аналогия в ежедневието, за да бъде обяснено (макар традиционно да се използват няколко примера – матрьошки, огледала и др.);
- изисква изграждането на умение у студентите да изпълняват рекурсивна функция мислено (т.е. „на ръка“) - съществен елемент от знанието за рекурсията според някои автори (Derek Wilcocks, 1994), (SEGAL, 1994);
- студентите вече имат изградени умения за организиране на повторения чрез итерация (оператор за цикъл) (Turbak, 1999);
- често от студентите се изисква да пишат рекурсивни програми, преди да са разбрали напълно поведението им.

Трудностите при изучаването и преподаването на рекурсия са мотивирали много изследователи да създадат визуални инструменти, подпомагащи разбирането ѝ.

Съществуващи автоматизирани средства за визуализация на изпълнението на рекурсивна функция

Универсални инструменти за визуализация на код

Python Tutor (Guo, 2013)

Python Tutor е един от най-популярните и широко използвани инструменти за визуализация на код. Това е уеб-базирано приложение, което поддържа множество програмни езици, включително Python, JavaScript, Java, C, C++ и други. Основната му функция е да позволи интерактивно, стъпка по стъпка изпълнение на код, като визуализира текущото състояние на програмата. За рекурсивни функции, Python Tutor предоставя ясна визуализация на стека от извикванията, показвайки как всяко рекурсивно извикване добавя нов кадър към стека и как стекът се "разгражда" при достигане на базовия случай. Това позволява на студентите ясно да видят дълбочината на рекурсията и потока на данни между различните нива на извикване. Инструментът е особено полезен за начинаещи, тъй като не изисква инсталация и може да се използва директно в уеб браузър, което го прави лесно достъпен за широка аудитория.

Recursion Tree Visualizer (Papa, 2020)

Освен Python Tutor, съществуват и други онлайн инструменти, които предоставят визуализация на рекурсия за множество програмни езици. Един пример е Recursion Tree Visualizer, който поддържа JavaScript, Python и Golang. Този инструмент е специализиран във визуализирането на рекурсивни дървета и позволява на потребителите да въвеждат рекурсивна функция или да използват предефинирани примери и да виждат как се изгражда дървото на извикванията в реално време. Той предоставя интерактивни функции, които позволяват анимационен и постъпков режим, промяна на входните данни на рекурсивната функция, както и решение с мемоизация.

Специализирани платформи за визуализация на алгоритми

VisuAlgo (Halim, 2011)

VisuAlgo е специализирана платформа за визуализация на богат набор от алгоритми. Тя е особено силна в областта на визуализирането на рекурсивни алгоритми като предоставя възможност за изследване на три различни типа рекурсивни функции – с едно, с две и с много рекурсивни извиквания. Освен това тя е полезна за преподаване на алгоритми за динамично програмиране, тъй като визуализира освен рекурсивното дърво и рекурсивния граф на изпълнението. Потребителите могат да въвеждат и собствен JavaScript код, за да генерират стъпка по стъпка рекурсивни дървета/графи, които визуализират изпълнението на алгоритъма.

ChiQat-Tutor (AlZoubi, 2020)

ChiQat-Tutor е модулна интелигентна обучителна система, предназначена да улесни изучаването на основни структури от данни и алгоритмични стратегии в компютърните науки, като един от ключовите модули е посветен на рекурсията. Тя интегрира множество визуализации и интерактивни дейности, за да отговори на предизвикателствата, с които се сблъскват обучаемите. Основният подход е базиран на визуалния модел „граф на рекурсията“ (RGraph) (Sa, 2010). RGraph е подобен на рекурсивните дървета, по това, че показва отделните рекурсивните извиквания, но добавя детайли в рекурсивните нива, като междинните резултати за всяко отложено изчисление. Основната разлика обаче между RGraph и рекурсивно дърво е, че рекурсивното дърво е абстрактна концепция, докато RGraph е проследим, тъй като показва последователността на извикване от едно към друго рекурсивно ниво. ChiQat-Tutor представлява по-напреднал подход, който комбинира визуализация с интелигентна обратна връзка и педагогически проектирани упражнения, които насърчават активното учене, включително отговаряне на въпроси, анимация, трасировка, валидация и конструкция на RGraph.

Езиково-специфични библиотеки и инструменти

SRec (Velázquez-Iturbide, 2016)

SRec (System for Recursion Animation) е ранен инструмент, разработен специално за визуализация на рекурсивна функция на Java. SRec поддържа три основни визуални представяния на рекурсията: трасировка на код, рекурсивен стек и рекурсивни дървета. В допълнение, той поддържа някои представяния, специфични за техниките за дизайн на алгоритми „разделяй и владей“ и динамично програмиране. Освен това, SRec предоставя цялостен набор от интерактивни операции насочени към анимиране на изпълнението на алгоритъм, преглеждане на визуализация, промяна на свойствата на графичните компоненти, филтриране на данните за показване, избиране или търсене на конкретни данни във визуализация, промяна на относителния ред на данните или обобщаване на визуализацията със статистика.

Обобщение и заключение

Анализът на различните налични инструменти показва, че те предлагат широк спектър от подходи за визуализиране на рекурсия, вариращи от общи визуализации на рекурсивни дървета и проследяване на стека, до специализирани инструменти за визуализиране на рекурсивни техники като динамично програмиране и „разделяй и владей“. Всеки инструмент има своите уникални предимства като достъпност, поддръжка на множество езици, визуализиране на сложни алгоритми и техники за дизайн на алгоритми, висококачествени анимации, но с изключение на SRec, всички те не визуализират графично достатъчно детайлно кода на функцията на всяко нейно рекурсивно ниво (екземплярите/инстанциите на функцията), което е от съществено значение за създаване на визуален модел на поведението на една рекурсивна функция.

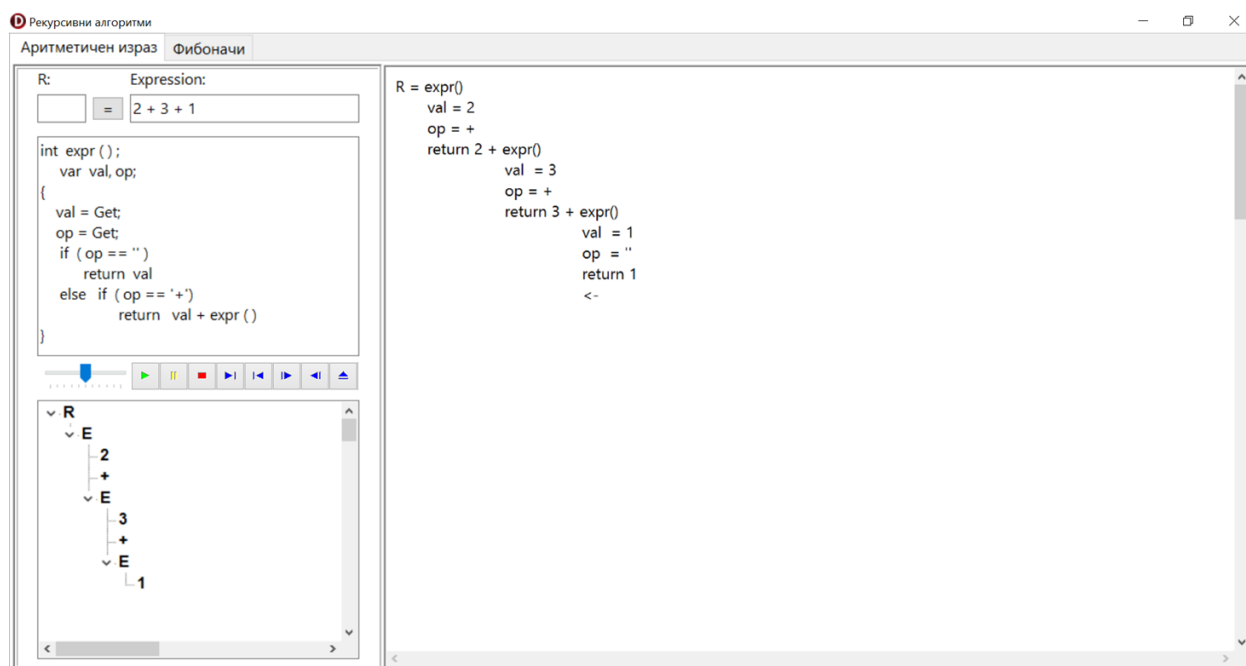
Предлагано авторско решение

Основните функционалности на разработения образователен инструмент са:

- визуализация на поведението на различни типове рекурсивна функция:
 - с единично рекурсивно извикване – линейна рекурсия;
 - с две рекурсивни извиквания в аритметичен израз;
 - с рекурсивно извикване в цикъл (например Backtracking) – предстои разработка;
- фокус върху постъпковото изпълнение на кода – с ясно представяне на кода, изпълнен при „потъване“ в рекурсията (активен/прав поток на управление), момента на връщане на предходно рекурсивно ниво и частта от кода, изпълняван при „изплуването“ от рекурсията (пасивен/обратен поток на управление);
- проследяване изменението на входни и входно-изходни параметри, както и на стойността на функцията;
- визуализация на строенето на рекурсивното дърво и рекурсивния стек;
- анимационен и постъпков режим на визуализацията с подходящи контроли.

Рекурсивна функция с единично рекурсивно извикване (линейна рекурсия)

Фигура 1 е екранна снимка на разработеното средство за визуализация на поведението на рекурсивна функция с единично извикване. Представител на този вариант на рекурсивна функция е изчисляването на аритметичен израз.

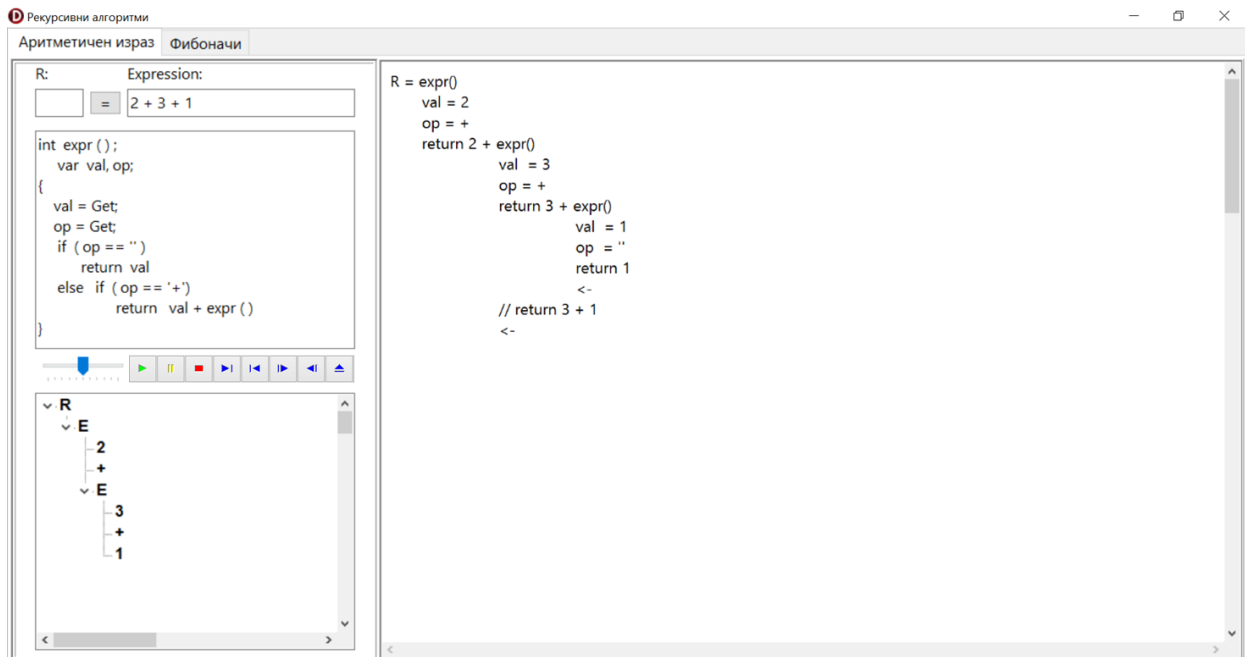


Фиг. 1. Визуализация на рекурсивна функция за изчисляване на аритметичен израз

Системата има 3 основни прозореца – за псевдокода на функцията, за трасировката му и за рекурсивното дърво. Визуализацията на трасировката на кода е с ясно разграничаване на отделните рекурсивни нива, изпълнените до момента оператори от кода за съответното рекурсивно ниво, както и момента на връщане към предходно рекурсивно ниво. Осигурени са контроли за анимацията и трасировката, като освен стандартните за стъпка напред/назад, има добавена и възможност за директно, а не постъпково изпълнение на дадено рекурсивно извикване (step over).

Представената екранна снимка на Фиг. 2 показва връщането на стойност на функцията на по-горно рекурсивно ниво и извършване на така наречените „отложени изчисления“,

отбелязани с //. В този момент рекурсивното дърво се „съкращава“, т.е. се превръща в рекурсивен стек.



Фиг. 2. Визуализация на извършването на „отложено изчисление“

Рекурсивна функция с две рекурсивни извиквания (разклонена рекурсия)

Като начален представител на разклонена рекурсия бе избрана рекурсивната функция за генериране на числата на Фибоначи. Визуализация на изпълнението ѝ е показано на Фиг. 3. Основният проблем при разклонената рекурсия е как да се визуализира графично йерархичното изпълнение на кода. Избраният подход с повторна визуализация на отложените изчисления (отбелязани с //) се оказа решение на този проблем.

ЗАКЛЮЧЕНИЕ

Докладът представя специално разработеното софтуерно средство, което използва визуална нотация за изпълнение на рекурсивна функция. То интерактивно демонстрира различни типове рекурсивни функции: с едно рекурсивно извикване (линейна рекурсия) и с две рекурсивни извиквания в аритметичен израз. Предстои разработка и на модул за визуализация на поведението на рекурсивна функция с многократно извикване в цикъл (напр. алгоритъм Backtracking).

Обучаемите могат да следят процеса на изпълнение стъпка по стъпка, както и изграждането на рекурсивното дърво/стек. Предимство на представената разработка е визуализацията на кода на отделните екземпляри на функцията на всяко рекурсивно ниво. Това спомага за създаването на така наречения „модел на копията“, считан за най-подходящия ментален модел на рекурсивния поток на управление на изчислението.

Системата е в начален етап на разработка и могат да бъдат посочени следните перспективи за развитие: визуализация на поведението на рекурсивна функция с входно-изходни параметри, както и на рекурсивна функция за работа с динамични структури от данни. Добре би било и добавянето на повече интерактивност, като даване на възможност за въвеждане на потребителска функция на някакъв програмен език.

