

В.І. Шинкаренко, Р.Р. Чигир

ІНСТРУМЕНТАЛЬНІ ЗАСОБИ КОНСТРУКТИВНО-ПРОДУКЦІЙНОГО МОДЕЛЮВАННЯ

Представлена концепція конструктивно-продукційного моделювання. Викладені основні принципи. Представлена класифікація конструкторів за метою конструювання та зовнішніми зв'язками. Визначені типи конструкторів: породжуючий, трансформуючий, аналізуючий, оптимізуючий/адаптуючий, алгоритмічний; автономний, параметричний, інтерактивний, мультиконструктор. Представлені досягнення із застосування конструктивно-продукційного підходу у рішенні низки задач. Для формування конструкторів розроблено інструментальне програмне середовище «Конструктор 1.0» засобами мови Python із використанням технології Qt для забезпечення кросплатформеності для формування конструкторів. На прикладі формування геометричних фракталів продемонстровані його функціональні можливості. Насамперед це стосується формування таких конструкторів як автономний породжуючий, параметричний трансформуючий, та об'єднуючий їх мультиконструктор. Показані особливості уточнюючих перетворень при формуванні конструкторів: спеціалізації, інтерпретації та конкретизації. Спеціалізація конструкторів визначає предметну область конструювання, необхідні дані і операції. Для забезпечення функціонування процесів конструювання всі операції конструкторів повинні інтерпретуватись відповідними процедурами алгоритмічного конструктору. Поєднанням конструктора (моделі елементів і можливих операцій) з алгоритмічним (моделі виконавця) формується конструктивна система, здатна автономно виконувати конструювання внутрішнім виконавцем. У процесі конкретизації задаються правила підстановки та початкові умови. Розроблене програмне середовище забезпечує певну гнучкість щодо можливих модифікацій конструкторів і процесів конструювання. Розроблений інструментарій може бути основою для моделювання різних конструкцій і конструктивних процесів, особливо у задачах їх оптимізації та структурної адаптації.

Ключові слова: конструктивно-продукційне моделювання, програмне забезпечення, конструктор, алгоритм, формальна граматики, формалізація

V.I.Shynkarenko, R.R.Chyhir

THE SOFTWARE TOOL OF CONSTRUCTIVE-SYNTHESIZING MODELLING

The concept of constructive-synthesizing modelling is presented. The basic principles are outlined. The classification of constructors by the purpose of constructing and external relations is presented. The types of constructors are defined: generating, transforming, analyzing, optimizing/adapting, algorithmic; standalone, parametric, interactive, multi-designer. Achievements in the application of the constructive-synthesizing approach to solving a number of problems are presented. The tool software environment «Constructor 1.0» has been developed for the formation of constructors by means of the Python language using Qt technology to ensure cross-platform compatibility. for the formation of constructors. On the example of the geometric fractal formation, its functionality is demonstrated. First of all, it concerns the formation of such constructors as a standalone generating, parametric transforming, and unifying multiconstructor. The features of expanding transformations in the formation of constructors are shown: specification, interpretation and concretization. The specialization of constructors determines the subject area of construction, the necessary data and operations. To ensure the functioning of the constructing processes, all constructor operations must be interpreted by the corresponding procedures of the algorithmic constructor. The combination of the constructor (model of elements and possible operations) with the algorithmic constructor (model of the executor) forms a constructive system capable of autonomously performing constructing by an internal executor. In concretization, substitution rules and initial conditions are specified. The developed software environment provides a certain flexibility in terms of possible modifications of constructors and constructing processes. The developed tools can be the basis for modelling various structures and constructing processes, especially in the tasks of their optimization and structural adaptation.

Key words: constructive-synthesizing modeling, software, constructor, algorithm, formal grammar, formalization

Вступ

Розглянемо конструктивну парадигму сприйняття світу людиною і відповідний підхід до програмування.

Основні положення конструктивної парадигми:

- світ сприймається як сукупність конструкцій і конструктивних процесів;
- конструкції складаються з деяких елементів і інших конструкцій;
- конструктивний процес складається з елементарних дій (дискретних або неперервних, детермінованих або стохастичних тощо) та інших процесів;
- елементи, конструкції, проміжні форми конструювання пов'язані між собою деякими відношеннями зв'язування;
- елементи конструкцій, проміжні форми конструювання, конструкції, відношення та операції – кожен має свій набір деякий атрибутів;
- у формуванні конструкцій використовується тріада: відношення $\rightarrow$ операція $\rightarrow$ відношення.

Останнє положення потребує деяких пояснень. У алгебрі, наприклад, цілих чисел з заданою операцією додавання, запис «3+4» сприймається як відношення: треба скласти 3 і 4, в результаті операції складання отримаємо результат 7. Тобто маємо відношення $\rightarrow$ операція: треба додати і додавання.

У конструктивній парадигмі: маємо відношення «гвинт можна закрутити у певний отвір» (водночас повинні співпадати деякі їх атрибути: розмір отвору і гвинта, вид різьби, шаг різьби). Це відношення є основою операції закручування гвинта в отвір. У результаті операції встановлюється нове відношення «закручений» у конструкцію з отвором гвинт. Відношення між отвором і гвинтом змінилось за тріадою пов'язаних між собою відношень і операції: можна закрутити $\rightarrow$ операція закручування $\rightarrow$ закручено. У результаті операції залишаються всі аргументи (елементи) з новим відношенням.

Парадигма конструкційно-продукційного моделювання з'явилася у результаті невдалої спроби узагальнення відомих

засобів формальних граматики. Як з'ясувалось, існує велика кількість модифікацій граматики [1, 2], таких як мультисимвольні [3], програмні, стохастичні [4, 5], зважені, індексні [6], матричні [7], різні графічні [5] та граматикиподібні L-системи [8], R-системи [9] тощо.

Основні переваги запропонованого підходу:

- засобами однієї моделі можна пов'язати конструкції з конструктивними процесами;
- оптимізувати або адаптувати структуру конструкції і процесів (складові, їхні розташування, зв'язки);
- моделювати конструкції, які складно, або навіть неможливо отримати іншими засобами;
- виконувати співставлення конструкцій;
- моделювати, аналізувати і прогнозувати конструктивні процеси.

У представленій роботі розглянуті функціональні можливості розробленого авторами інструментального середовища «Конструктор 1.0» для автоматизації розробки конструктивно-продукційних моделей та їх використання.

Пов'язані роботи

Виконана низка досліджень з використанням конструктивно-продукційного моделювання, які засвідчують продуктивність та ефективність цього підходу. Наведемо деякі з них.

Виявлені нові можливості формування геометричних фракталів [10]. А саме: формування фракталів з неоднорідних елементів; комбінування різних, у тому числі класичних, фракталів у мультифракталах. Можливості формування фракталів розширені шляхом усунення необхідних за інших підходів обмежень: елементи формування фракталів повинні бути представлені істотно непересічними множинами; застосовуються лише стискаючі відображення.

Структурну адаптацію алгоритмів і процесу стискання даних формалізовано сукупністю трьох конструкторів [11]: двоїсто-

го конструктора алгоритму стиснення (компресії та декомпресії), перетворювача сконструйованого алгоритму в конструктивний процес стиснення і безпосередньо конструктора-адаптера. Реалізацією конструкторів є відповідне програмне забезпечення. Встановлена його функціональна ефективність під час стискання різних банків даних.

Розроблена конструкційно-продукційна модель природної мови [12] дозволяє: розглядати мову як конструктивний процес, що може бути використаний як основа для створення методології побудови систем з високим ступенем інтелектуальності; удосконалити процеси семантичного аналізу, зокрема, у задачах зіставлення та виявлення відповідного семантичного змісту в текстах. Таким чином суттєво зменшується вплив синонімів, омонімів, перифразів та перекладу.

Моделювання ґраток кристалів для дослідження існуючих і моделювання нових матеріалів, вивчення процесів кристалізації. Для моделювання кристалічних ґраток застосоване конструктивно-продукційне моделювання [13]. Конструктори враховують атрибути елементної бази, що дозволяє природним чином будувати фрактальні просторові графи за правилами підстановки. Кристалічні ґратки розглядаються як часткові випадки фрактальних просторових графів.

Використання моделювання у формуванні грозових розрядів на основі конструктивно-продукційного підходу дозволяє отримати реалістичний опис грозової фронтальної активності [14]. Такий підхід може бути покладено в основу вирішення динамічної задачі захисту від блискавок інженерних споруд та цивільних об'єктів, а також розробки стратегії поведінки літальних апаратів з метою мінімізації ризиків ураження блискавкою в умовах руху в грозовому фронті.

Застосування засобів конструктивно-продукційного моделювання дозволило формалізувати процеси і результат проектування структур на логічному рівні [15]. Реалізація конструктора представляє моделі логічно пов'язаних конкретних елементів даних. Розроблена також конструктивна модель представлення і адаптації да-

них у оперативній пам'яті за логічною моделлю.

Для рішення кожної з цих і багатьох інших задач конструювання і моделювання (кодування і налагодження комп'ютерних програм, методу аналізу ієрархій, виявлення запозичень у текстах, визначення авторського стилю і автора тексту, фрактальних часових рядів, розкладу занять університету, формування онтологій залізничного транспорту) розроблялось спеціальне програмне забезпечення. Задача даної роботи – спростити конструктивне моделювання шляхом розробки універсального програмного середовища.

Види конструкторів

Усі конструктори формуються на основі узагальненого конструктора, теоретичні засади якого закладені в [1] (де для конструктора використовувався застарілий термін – формальна структура).

Узагальнений конструктор (абстрактний) визначається як:

$$C = \langle M, \Sigma, \Lambda \rangle,$$

де M – поновлюваний неоднорідний носій, Σ – сигнатура відносин та пов'язаних з ними операцій, Λ – інформаційне забезпечення конструювання: призначення, умови початку та завершення конструювання, правила підстановки та обмеження.

Формування реальних конструкторів виконується уточнюючими перетвореннями узагальненого: конкретизації, спеціалізації, інтерпретації та реалізації. Їх сутність покажемо далі на прикладах.

За метою використання конструктори поділяються на:

- породжуючі, які заданими правилами виконують формування конструкцій/конструктивних процесів (однієї або декількох);

- трансформуючі, які за однією конструкцією (її моделлю) формують іншу. Наприклад, програму у кодах процесору у конструктивний процес її виконання;

- аналізуючі, які виконують аналіз конструкцій. Наприклад, перевіряти чи може конструкція бути сформована за наявними правилами;

– оптимізуючі/адаптуючі, які здатні змінювати структуру конструкції (складові, їх порядок та зв'язки);

– алгоритмічні – в поєднанні з іншими конструкторами створюють конструктивну систему: модель елементів, проміжних форм і можливих операцій над ними поєднується з моделлю внутрішнього виконавця, який вмiє виконувати всі операції.

За зовнішніми зв'язками виділяються конструктори:

– автономні, у яких є вся необхідна інформація для досягнення мети і для цього не потрібна інформація ззовні;

– параметричні, яким надаються зовнішні дані перед початком роботи через механізм параметрів;

– інтерактивні, які мають операції отримання даних ззовні;

– мультиконструктори, які складаються з декількох поєднаних конструкторів.

Функціональність «Конструктору 1.0»

Програмне середовище «Конструктор 1.0» розроблено засобами мови Python із використанням технології Qt для забезпечення кросплатформеності.

Продемонструємо його функціональність на прикладі моделювання геометричного фракталу «Сніжинка Коха» [16].

На головному екрані додаємо до списку конструкторів по чергово конструктори (рис. 1). Кожен з них наслідуює всі властивості узагальненого конструктора.



Рис. 1. Перелік конструкторів

Спочатку додамо мультисимвольний автономний конструктор.

Перейдемо його уточнення на наступному екрані (рис.2).



Рис. 2. Уточнюючі перетворення

У процесі спеціалізації задається предметна область, опис потрібних даних, елементів, відношень і операцій. У цьому конструкторі додаткових операцій не передбачено, дані задаються як на рис.3. Без спеціалізації, інтерпретації та конкретизації реалізація не доступна (не можлива).

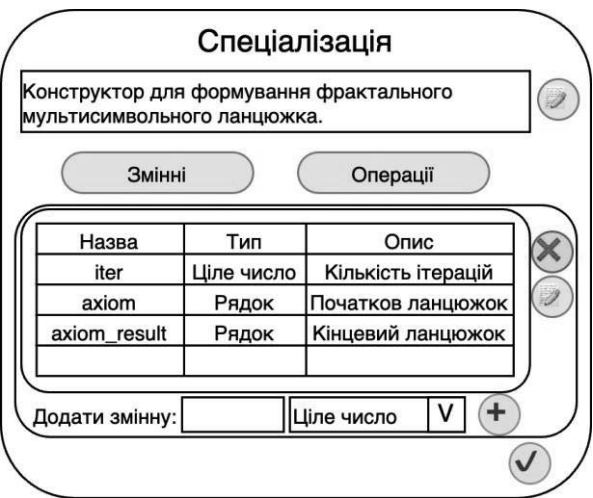


Рис.3. Спеціалізація мультисимвольного конструктора

Для інтерпретації мультисимвольного конструктора потрібен відповідний алгоритмічний конструктор. Задаємо його у списку конструкторів. На рис.1 він названий як «L-системний». Його спеціалізація (рис. 4) полягає у наданні налагоджених процедур, які спроможні виконувати всі операції мультисимвольного конструювання.

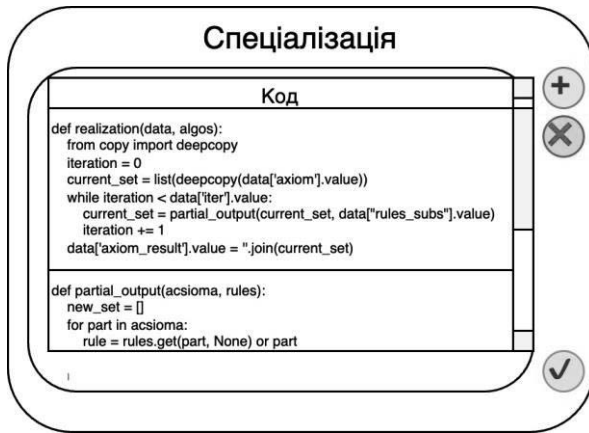


Рис. 4. Спеціалізація алгоритмічного конструктора

Тепер можна повернутись до інтерпретації мультисимвольного конструктора (рис. 5).

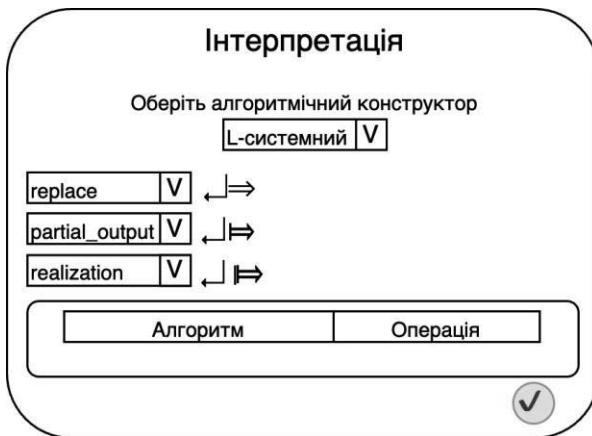


Рис.5. Інтерпретація мультисимвольного конструктора

У процесі інтерпретації кожній допустимій операції мультисимвольного конструктора (вказаній під час його спеціалізації) ставиться у відповідність алгоритм алгоритмічного конструктора. Обов'язково повинні інтерпретуватись операції підстановки, часткового і повного виводу, які успадковуються від узагальненого конструктора [1].



Рис.6. Встановлення значень змінних під час конкретизації

У результаті інтерпретації формується конструктивна система яка поєднує конструктор з елементами і можливими операціями та модель внутрішнього виконання, здатного ці операції виконувати.

Конкретизація (рис. 6 і 7) задає правила підстановки (конструювання), які складаються з відношень підстановки і операцій над атрибутами та початкові умови формування.

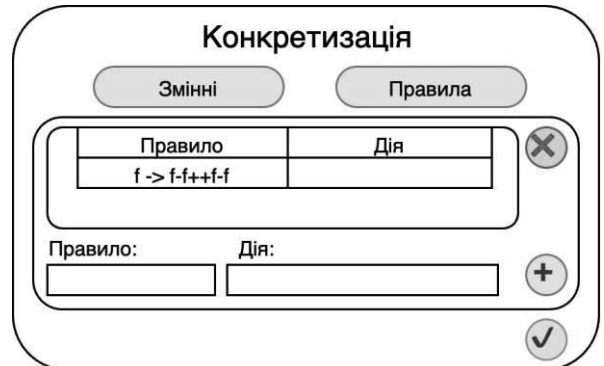


Рис. 7. Додавання правил підстановки під час конкретизації

Так як мультисимвольний конструктор автономний, то після всіх уточнюючих перетворень стає можливою його реалізація, тобто формування фрактальної мультисимвольної послідовності за правилами формування «сніжинки Коха». Мультисимвольна послідовність буде збережена у заданий файл.

Для відображення за цією послідовністю геометричного фракталу розробляється параметричний конструктор, параметром якого буде конструкція, сформована мультисимвольним конструктором.

Задаємо параметричний конструктор відповідно до рис. 1.

Він спеціалізується як на рис. 8 і 9. У цьому конструкторі вже задані деякі операції.

Аналогічно з мультисимвольним конструктором, для параметричного задається відповідний алгоритмічний конструктор з відображення фракталу і виконується інтерпретація, як на рис. 10 (на відміну від першого конструктора є додаткові інтерпретовані операції, пов'язані з відрисовкою ліній та зміною поточного кута нахилу).

Спеціалізація

Конструктор для формування графічного відображення.

Змінні

Операції

Назва	Тип	Опис
current_angle	Дробове число	Поточне значення кута
axiom	Рядок	Початков ланцюжок
disp_len	Дробове число	Знач. дисперсії довжини
mat_oz_len	Дробове число	Знач. мат. очікування дов
disp_angle	Дробове число	Знач. дисперсії кута
mat_oz_angle	Дробове число	Знач. мат. очікування кута

Додати змінну: Ціле число V

Рис. 8. Завдання змінних під час спеціалізації графічного конструктора

Спеціалізація

Конструктор для формування графічного відображення.

Змінні

Операції

Операція	Параметри	Опис
v	(val, len, angle)	Провести лінію
-	(val, step, mat, disp)	Зменшити кут нахилу
+	(val, step, mat, disp)	Збільшити кут нахилу

Додати операцію: з параметрами

Рис. 9. Завдання операцій під час спеціалізації графічного конструктора

Інтерпретація

Оберіть алгоритмічний конструктор

Відображення фракталу ☒

replace ☒

partial_output ☒

realization ☒

Алгоритм	Операція
use_f	V
use_minus	-
use_plus	+

Рис. 10. Інтерпретація операцій графічного конструктора

Конкретизація графічного конструктора наведена на рис. 11 і 12

Конкретизація

Змінні

Правила

Назва	Значення
current_angle	60.0
axiom	A
disp_len	0.0
mat_oz_len	1.0
disp_angle	0.0
mat_oz_angle	60.0

Рис. 11. Ініціалізація змінних графічного конструктора

Конкретизація

Змінні

Правила

Правило	Дія
A -> fA, A -> vA	v(current_x, current_y, len, angle)
A -> -A	-(current_len, step_len, mat_oz_len, disp_len); -(current_angle, step_angle, mat_oz_angle, disp_angle)
A -> +A	+(current_len, step_len, mat_oz_len, disp_len); +(current_angle, step_angle, mat_oz_angle, disp_angle)
A -> ε	

Правило: Дія:

Рис. 12. Завдання правил підстановки графічного конструктора

Виконати «Графічний» конструктор (етап реалізації) неможливо автономно. Він може працювати лише у зв’язці з мультисимвольним. Тому задається мультиконструктор «Сніжинка Коха». Він спеціалізується як на рис. 13 і рис. 14. Спочатку вказується як мультисимвольний і графічний конструктор поєднуються (рис.13), а потім які дії слід виконати для повного формування зображення сніжинки Коха.

Так як «Графічний» конструктор є трансформуючим, задається два відношення підстановки: перше використовується

для розбору вхідної конструкції (мультисимвольного ланцюжка), друге – для формування вихідної конструкції (зображення сніжинки Коха).

Задані операції над атрибутами змінюють поточний кут нахилу лінії з можливими стохастичними збуреннями.

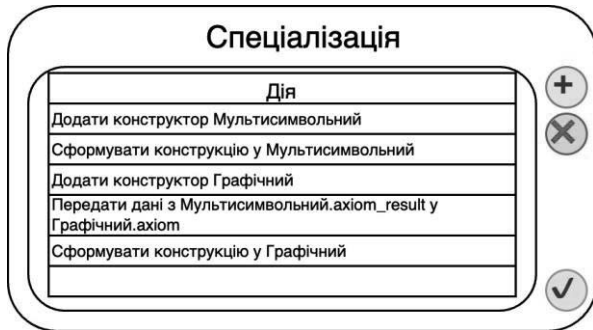


Рис. 13. Порядок виконання мультисимвольного конструктора



Рис. 14. Встановлення зв'язку між конструкторами

У процесі реалізації мультиконструктора отримаємо файл із зображенням сніжинки Коха (рис. 15).

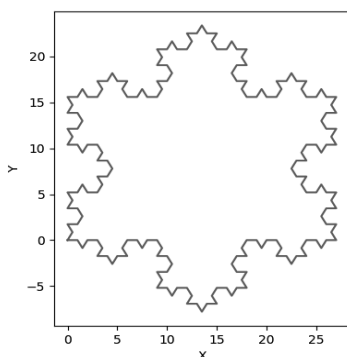


Рис. 15. Сніжинка Коха

Розроблене інструментальне середовище є максимально гнучким щодо можливих модифікацій. Так, наприклад, завдання ненульової дисперсії дозволить формувати сніжинку Коха з певними стохастичними збуреннями. Зміна аксіом і правил у мультисимвольному конструкторі дає можливість формувати інші відомі і нові геометричні фрактали, зміна процедури v у алгоритмічному конструкторі відображення фракталів дозволяє іншим чином рисувати лінії (змінюючи колір, передуючі кольори, обрати інший тип ліній, замінивши лінію деяким зображенням тощо).

Висновки

Конструктивно-продукційне моделювання є особливо корисним у випадку рішення задач оптимізації та структурної адаптації конструкцій та конструктивних процесів.

У роботі представлений інструментарій для автоматизації формування конструктивно-продукційних моделей об'єктів та процесів різної природи.

Наведено опис застосування цього інструментарію для розробки мультиконструктора та його використання на прикладі формування та відображення геометричних фракталів.

У процесі розробки представленого інструментарію враховано та узагальнено досвід розробки більше 20 конструктивно-продукційних моделей, програмна реалізація яких була індивідуальною для кожної з них.

Застосування універсального інструментарію дозволить систематизувати процес розробки моделей та значно скоротити час їх формування.

З метою подальшої розробки конструкторів для інших предметних областей можна формувати конструкції і конструктивні процеси іншої природи з використанням уніфікованого інструментарію.

Література

1. V.I. Shynkarenko, V.M. Ilman, Constructive-Synthesizing Structures and Their Grammatical Interpretations.; Part I. Generalized Formal Constructive-Synthesizing Structure, *Cybernetics and Systems Analysis*, 2014.

- № 50(5), pp.665–662. doi: 10.1007/s10559-014-9655-z; Part II. Refining Transformations, *Cybernetics and Systems Analysis*, 2014. № 50(6). pp.829–841. doi: 10.1007/s10559-014-9674-9.
2. S. Müller, Grammatical theory: From transformational grammar to constraint-based approaches, *Language Science Press*, 2023. doi: 10.5281/zenodo.3992307.
3. O. Iwashokun, A. Ade-Ibijola, Parsing of Research Documents into XML Using Formal Grammars, *Applied Computational Intelligence and Soft Computing*, 2024. doi: 10.1155/2024/6671359.
4. R. Wandr, A.L.S. Ferreira, R.V.S. Pessoa, A. Galv, A. Machado-Lima, A stochastic grammar approach to mass classification in mammograms, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.
5. X. Pu, M. Kay, A probabilistic grammar of graphics. *CHI Conference on Human Factors in Computing Systems*, 2020. pp.1–13. doi: 10.1109/TCBB.2023.3247144.
6. F. D'Alessandro, O.H. Ibarra, I. McQuillan, On finite-index indexed grammars and their restrictions, *International Conference on Language and Automata Theory and Application*, 2021. №279. C.287–298. doi: 10.1016/j.ic.2020.104613.
7. P. Prusinkiewicz, A. Lindenmayer, The algorithmic beauty of plants, *Springer Science & Business Media*, 2004. pp.228. doi: 10.1007/978-1-4613-8476-2.
8. L.P. Lisovik, T.A. Karnaukh, A method of specification of fractal sets, *Cybernetics and Systems Analysis*, 2009. № 45(3). pp.365–372. doi: 10.1007/s10559-009-9117-1.
9. V.I. Shynkarenko, Constructive-Synthesizing Representation of Geometric Fractals, *Cybernetics and Systems Analysis*, 2019. № 55. pp.186-199. doi: 10.1007/s10559-019-00123-w
10. V.I. Shynkarenko, T.M. Vasetska, Modeling the Adaptation of Compression Algorithms by Means of Constructive-Synthesizing Structures, *Cybernetics and Systems Analysis*, 2015. № 51(6). pp.849–861. doi: 10.1007/s10559-015-9778-x.
11. V. Shynkarenko, O. Kuropiatnyk, Constructive Model of the Natural Language, *Acta Cybernetica*, 2018. № 23(4). C.995–1015. doi: 10.14232/actacyb.23.4.2018.2.
12. V. Shynkarenko, O. Letuchyi, R. Chyhir, Constructive-synthesizing modeling of fractal crystal lattices, *18th IEEE International Conference on Computer Science and Information Technologies (CSIT)*, 2023. doi:10.1109/CSIT61576.2023.10324251
13. V. Shynkarenko, K. Lytvynenko, R. Chyhir, I. Nikitina, Modeling of Lightning Flashes in Thunderstorm Front by Constructive Production of Fractal Time Series, *Advances in Intelligent Systems and Computing IV*, 2020. № 1080. pp.173–185. doi: 10.1007/978-3-030-33695-0_13
14. В.И. Шинкаренко, В.М. Ильман, Г.В. Забула, Конструкционно-продукционная модель структур данных на логическом уровне, *Проблемы програмування*, 2014. № 2-3. С. 10–16.
15. J.W. Harris, H. Stocker, Handbook of Mathematics and Computational Science. *Springer-Verlag*, New York, 1998. pp.114–115.

References

1. V.I. Shynkarenko, V.M. Ilman, Constructive-Synthesizing Structures and Their Grammatical Interpretations.; Part I. Generalized Formal Constructive-Synthesizing Structure, in: *Cybernetics and Systems Analysis*, 2014. № 50(5), pp.665–662. doi: 10.1007/s10559-014-9655-z; Part II. Refining Transformations, in: *Cybernetics and Systems Analysis*, 2014. № 50(6). C.829–841. doi: 10.1007/s10559-014-9674-9.
2. S. Müller, Grammatical theory: From transformational grammar to constraint-based approaches, *Language Science Press*, 2023. doi: 10.5281/zenodo.3992307.
3. O. Iwashokun, A. Ade-Ibijola, Parsing of Research Documents into XML Using Formal Grammars, in: *Applied Computational Intelligence and Soft Computing*, 2024. doi: 10.1155/2024/6671359.
4. R. Wandr, A.L.S. Ferreira, R.V.S. Pessoa, A. Galv, A. Machado-Lima, A stochastic grammar approach to mass classification in mammograms, in: *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2023.
5. X. Pu, M. Kay, A probabilistic grammar of graphics, in: *CHI Conference on Human Factors in Computing Systems*, 2020. pp.1–13. doi: 10.1109/TCBB.2023.3247144.
6. F. D'Alessandro, O.H. Ibarra, I. McQuillan, On finite-index indexed grammars and their restrictions, in: *International Conference on Language and Automata Theory and Application*, 2021. №279. pp.287–298. doi: 10.1016/j.ic.2020.104613.

7. P. Prusinkiewicz, A. Lindenmayer, The algorithmic beauty of plants, Springer Science & Business Media, 2004. pp.228. doi: 10.1007/978-1-4613-8476-2.
8. L.P. Lisovik, T.A. Karnaukh, A method of specification of fractal sets, in: Cybernetics and Systems Analysis, 2009. № 45(3). pp.365–372. doi: 10.1007/s10559-009-9117-1.
9. V.I. Shynkarenko, Constructive-Synthesizing Representation of Geometric Fractals, in: Cybernetics and Systems Analysis, 2019. № 55. pp.186-199. doi: 10.1007/s10559-019-00123-w
10. V.I. Shynkarenko, T.M. Vasetska, Modeling the Adaptation of Compression Algorithms by Means of Constructive-Synthesizing Structures, in: Cybernetics and Systems Analysis, 2015. № 51(6). pp.849–861. doi: 10.1007/s10559-015-9778-x.
11. V. Shynkarenko, O. Kuropiatnyk, Constructive Model of the Natural Language, in: Acta Cybernetica, 2018. № 23(4). pp.995–1015. doi: 10.14232/actacyb.23.4.2018.2.
12. V. Shynkarenko, O. Letuchyi, R. Chyhir, Constructive-synthesizing modeling of fractal crystal lattices, in: 18th IEEE International Conference on Computer Science and Information Technologies (CSIT), 2023. doi: 10.1109/CSIT61576.2023.10324251
13. V. Shynkarenko, K. Lytvynenko, R. Chyhir, I. Nikitina, Modeling of Lightning Flashes in Thunderstorm Front by Constructive Production of Fractal Time Series, in: Advances in Intelligent Systems and Computing IV, 2020. № 1080. pp.173–185. doi: 10.1007/978-3-030-33695-0_13
14. V.I. Shynkarenko, V.M. Ilman, G.V. Zabula, Constructive-synthesizing model of data structures at logical level, in: Problems of programming, 2014. № 2-3. pp. 10–16.
15. J.W. Harris, H. Stocker, Handbook of Mathematics and Computational Science. Springer-Verlag, New York, 1998. pp.114–115.

Одержано: 14.04.2024

Внутрішня рецензія отримана: 23.04.2024

Зовнішня рецензія отримана: 30.04.2024

Про авторів:

Шинкаренко Віктор Іванович,
доктор технічних наук, професор.
<https://orcid.org/0000-0001-8738-7225>.

Чигир Роберт Романович,
аспірант
<https://orcid.org/0000-0002-7439-7368>.

Місце роботи авторів:

Український державний університет
науки і технологій,
тел. +38-056-373-15-05
E-mail: office@ust.edu.ua
Веб-сайт: <https://ust.edu.ua>