

УДК 004.423

Киселевич В. В., асистент

Житомирський державний університет імені Івана Франка

ГІБРИДНИЙ МЕТОД СТАТИЧНОГО АНАЛІЗУ АСИМПТОТИЧНОЇ СКЛАДНОСТІ ПОТОКІВ ВИКОНАННЯ КОДУ

Оцінка продуктивності програмних систем здійснюється на двох рівнях, між якими існує суттєва прогалина. Інструменти профілювання та трасування виявляють вузькі місця лише, коли система працює під навантаженням – після розгортання. Натомість статичні аналізатори обчислювальної складності визначають складність окремих процедур, проте потребують повного доступу до вихідного коду всіх компонентів. Інструменти на основі великих мовних моделей (LLM) оцінюють складність довільного коду, проте без формальних гарантій коректності [1, 2]. Відтак бракує підходу, який дозволяв би ще на етапі розробки визначити асимптотичну складність цілого потоку виконання – послідовності операцій від точки входу бізнес-процесу до його завершення.

Метою роботи є розробка гібридного методу статичного аналізу асимптотичної складності потоків виконання програмного коду, який поєднує детерміністичний аналіз синтаксичних конструкцій із залученням LLM для семантично складних випадків.

Розглянемо потік формування фінансового звіту: система завантажує дані клієнта, отримує перелік замовлень, після чого для кожного замовлення виконує запит на отримання товарів та обчислює підсумок. Кожна операція окремо має прийнятну складність, однак композиція вкладених викликів утворює потік зі складністю $O(m \times k)$, де m – кількість замовлень, k – кількість товарів. Профілювання виявить це лише за великих обсягів даних, тоді як статичний аналіз потоку вкаже на проблему до першого запуску. У попередньому дослідженні [3] запропоновано застосування нотації O -велике на рівні модулів архітектури; дана робота розвиває цей підхід на рівні потоків виконання.

Запропонований гібридний метод ґрунтується на побудові дерева потоку виконання з абстрактного синтаксичного дерева аналізованої функції. Кожен вузол відповідає конструкції коду: послідовність, розгалуження, цикл, рекурсія або термінальна операція. Термінальні операції поділяються на детерміністичні (стандартні операції з відомою складністю) та семантичні (зовнішні виклики). Перші оцінюються

зіставленням із шаблонами з рівнем довіри 1,0; другі – за допомогою LLM із багаторазовим запуском та консенсусом, де рівень довіри визначається часткою узгоджених відповідей. Результати агрегуються від листків до кореня за правилами теорії складності [4]: для послідовних операцій обирається домінуючий компонент, для розгалужень – найгірший сценарій, для вкладених структур складності перемножуються. Числова оцінка довіри поширюється через усе дерево, кількісно виражаючи невизначеність результату.

Для валідації розроблено прототип мовою TypeScript та проведено тестування на 30 функціях із відомою складністю – від лінійного пошуку до рекурсивних алгоритмів. У режимі детерміністичного аналізу точність становить 100 % для структурних шаблонів та 80 % загалом. Серед оцінок із рівнем довіри $\geq 0,5$ точність становить 96 %: за достатньої впевненості прототип дає коректний результат, а за її відсутності позначає результат як невизначений. Решта випадків – функції, складність яких визначається семантикою алгоритму; для них передбачено компонент на основі LLM.

Запропонований підхід покликаний доповнити інструментарій аналізу коду, заповнюючи прогалину між статичним аналізом окремих алгоритмів та профілюванням у робочому середовищі. Серед обмежень: O-велике не враховує сталих множників; прототип підтримує лише TypeScript; тестовий набір потребує розширення. Подальші дослідження спрямовані на підтримку інших мов та інтеграцію в автоматизовану перевірку коду.

Список використаних джерел:

1. Sikka J., Satya K., Kumar Y. et al. Learning Based Methods for Code Runtime Complexity Prediction. *ECIR 2020*. Cham : Springer, 2020. P. 313–325.
2. Dube E., Parr T., Soares G. BigO(Bench): Can LLMs Generate Code with Controlled Time and Space Complexity. *arXiv preprint*. 2025. URL: <https://arxiv.org/abs/2503.15242>.
3. Киселевич В., Іванов Д. Використання нотації O-велике у розробці інформаційних систем. *Сучасні інформаційні технології в освіті та науці* : IX Всеукр. наук.-практ. конференція з міжнар. участю, 21–22 листопада 2024 р., Житомир, 2024. С. 375–379.
4. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. 4th ed. Cambridge : MIT Press, 2022. 1312 p.