

РЕЦЕНЗІЯ

на дисертаційну роботу

Форкерта Павла Павловича

на тему «**Дослідження використання рантайму Go як платформи для
побудови нових мов програмування**»,

яка представлена на здобуття ступеня доктора філософії

за спеціальністю 121 Інженерія програмного забезпечення

Актуальність теми дисертації. У сучасній мовній інженерії виразною є тенденція до проектування нових мов програмування, діалектів і DSL не з нуля, а на базі випробуваних платформ (JVM, LLVM, GraalVM/Truffle чи JavaScript-рушіїв). Такий підхід дає змогу делегувати хост-платформі задачі компіляції, керування пам'яттю та виконання коду, зосередивши ресурси на розробці фронтенду й семантики. Водночас потенціал Go як хост-середовища досі вивчений лише фрагментарно: у науково-технічній літературі Go позиціонується здебільшого як самостійний інструмент для хмарної та серверної розробки, а не як цільова платформа транспіляції. Рецензована робота присвячена саме цій прогалині – систематичному дослідженню того, для яких класів мов і мовних розширень середовище виконання (рантайм) мови Go є зручним та економічно виправданим фундаментом. Зважаючи на високу популярність Go та потребу індустрії в доступних інструментах створення DSL, обраний здобувачем напрям дослідження є безсумнівно актуальним як у теоретичному, так і в практичному аспектах.

Зв'язок роботи з науковими програмами, планами, темами. Дисертацію виконано на кафедрі інженерії програмного забезпечення та інформаційних технологій Дніпровського національного університету імені Олеся Гончара відповідно до індивідуального плану підготовки здобувача ступеня доктора філософії. Дослідження пов'язане з науково-дослідними роботами № ФПМ-2-22 «Розроблення програмного забезпечення аналізу та кластеризації часових рядів» (2022–2024 рр., номер держреєстрації 0122U001465) та № 58 – ФПМ-2-25

«Розроблення інформаційної технології обробки статистичних даних» (2025–2027 рр., номер держреєстрації 0125U002280).

Формулювання наукової задачі, нове вирішення якої одержане в дисертації, полягає в системному обґрунтуванні доцільності використання рантайму мови Go як хост-платформи для побудови нових мов програмування шляхом їх транспіляції у валідний Go-код, а також у розробленні й експериментальній апробації відповідних архітектурно-програмних засобів.

Для розв’язання окресленої задачі автором проведено критичний аналіз сучасних підходів до імплементації мов програмування на базі наявних платформ; досліджено інженерно-технічні властивості рантайму та інструментальної екосистеми Go; сформульовано комплексну систему критеріїв оцінювання придатності Go як цілі транспіляції; розроблено узагальнену багаторівневу архітектуру транспілятора; створено та випробувано експериментальний діалект GoNext, що містить репрезентативний набір мовних розширень; здійснено оцінювання отриманих результатів і їх подальше узагальнення на ширший клас предметно-орієнтованих мов та інженерних задач.

Об'єктом дослідження є процеси проектування та реалізації нових мов програмування і мовних розширень на базі наявних програмних платформ. *Предметом* дослідження виступають методи, моделі та програмні засоби використання рантайму Go як платформи для побудови нових мов програмування або діалектів існуючих.

Наукова новизна отриманих автором результатів. До найвагоміших результатів роботи, що визначають її наукову новизну, належать такі:

1. Вперше системно обґрунтовано розгляд рантайму Go як хост-платформи особливого типу — не багатомовної віртуальної машини і не низькорівневого компіляторного фреймворку, а цілі транспіляції для мов із високим рівнем інтероперабельності з екосистемою Go.

2. Удосконалено підхід до оцінювання придатності хост-платформи для транспіляції завдяки комплексному врахуванню виконувальних властивостей,

виразних можливостей хост-мови, інтероперабельності, інструментальної підтримки та вартості реалізації мовних конструкцій.

3. Удосконалено архітектурний підхід до побудови мовних розширень у формі надмножини Go з поетапною трансформацією у звичайний Go-код, який поєднує нормалізацію конструкцій, частковий семантичний аналіз і перетворення у вузли `go/ast`, що дозволяє коректно реалізовувати конструкції, залежні від сигнатур функцій, областей видимості та правил уніфікації типів Go.

4. Вперше в межах єдиного транспіляційного підходу продемонстровано реалізацію комплексу мовних розширень поверх рантайму Go без модифікації офіційного компілятора: повнофункціональних `enum`-типів, зіставлення із взірцем, іменованих аргументів, значень параметрів за замовчуванням, методів розширення, узагальнених методів, коротких анонімних функцій та універсального синтаксису виклику функцій.

Особливо цінною з теоретичного погляду є запропонована в другому розділі класифікація перетворень мовних конструкцій на синтаксичні, структурні та репрезентаційні, оскільки вона надає дослідникові інструмент апріорного прогнозування складності й накладних витрат реалізації нового мовного розширення ще до початку його втілення.

Практичне значення одержаних результатів. Практична цінність роботи не викликає сумнівів. Запропоновані підхід і архітектура дозволяють істотно знизити вартість створення нових мов, діалектів і предметно-орієнтованих мов за рахунок повторного використання компілятора, рантайму, системи збірки та бібліотек Go. Розроблена в четвертому розділі покрокова методика прийняття рішення про вибір Go як цільової платформи має самостійну інженерну цінність і може безпосередньо застосовуватися проєктувальниками нових мов і DSL. Експериментальний діалект GoNext підтверджує практичну здійсненність підходу та може слугувати відправною точкою для подальших прикладних розробок, зокрема мов опису конфігурацій, інтеграційних процесів, серверних предметно-орієнтованих мов, а також формальних мов гнучкого надання ресурсів вебслужб у мікросервісних архітектурах.

Обґрунтованість і достовірність наукових положень, висновків і рекомендацій, які захищаються. Достовірність результатів забезпечується коректним застосуванням методів теорії мов програмування та компіляторобудування, узгодженістю теоретичних положень другого розділу з експериментальними результатами третього, працездатністю створеного транспілятора GoNext, а також відповідністю отриманих висновків опублікованим результатам інших дослідників, зокрема щодо реалізації мов на платформах JVM, LLVM і Truffle. Висновки дисертації логічно впливають зі змісту роботи й не містять внутрішніх суперечностей.

Повнота викладу результатів дисертації в публікаціях. Основні результати дослідження опубліковано в 11 наукових працях, серед яких 3 статті у наукових фахових виданнях України (категорія Б) та 8 тез доповідей у матеріалах міжнародних і всеукраїнських наукових конференцій. Опубліковані праці повною мірою висвітлюють зміст дисертаційної роботи та відображають її основні теоретичні й практичні результати.

Особистий внесок здобувача. Усі основні наукові положення, теоретичні узагальнення та практичні результати дисертаційної роботи отримано автором особисто. У працях, опублікованих у співавторстві, здобувачеві належать дослідження придатності Go як хост-платформи, розроблення архітектури транспілятора, реалізація діалекту GoNext та аналіз схем трансформації мовних розширень.

Оцінка структури, змісту та оформлення дисертації. Робота викладена на 204 сторінках, складається зі вступу, чотирьох розділів, висновків, списку використаних джерел із 92 найменувань та додатків; обсяг основного тексту становить 116 сторінок. Перший розділ дає ґрунтовний огляд підходів до реалізації мов на базі наявних платформ і чітко окреслює дослідницьку нішу. Другий розділ формує теоретичне ядро роботи – систему критеріїв оцінювання хост-платформи, узагальнену архітектуру транспілятора, класифікацію трансформацій і принципи проектування нових мов на базі Go. Третій розділ присвячено експериментальній реалізації – діалекту GoNext; тут на конкретних

мовних розширеннях простежено весь шлях від синтаксису до згенерованого Go-коду, включно з аналізом альтернативних схем представлення перелічуваних типів (enum) та їх узгодженості зі збирачем сміття. Четвертий розділ містить комплексне оцінювання підходу, порівняння Go з альтернативними платформами, аналіз обмежень і ризиків та практичні рекомендації. Матеріал викладено послідовно, грамотною науковою українською мовою; текст добре ілюстровано прикладами вихідного та згенерованого коду, які суттєво полегшують сприйняття.

Зауваження та побажання до дисертаційної роботи.

1. Синтаксис нових мовних конструкцій GoNext і схеми їх трансформації у розділах 2 і 3 описано переважно на рівні ілюстративних прикладів коду та словесних алгоритмів; формальної граматики розширень у роботі не наведено. Бажано було б доповнити цей опис формальним поданням — граматикою нових конструкцій, наприклад у розширеній нотації Бекуса–Наура, та правилами переписування хоча б для окремих трансформацій.

2. У підрозділі 3.5 серед можливостей конструкції match заявлено перевірку вичерпності на етапі компіляції за відсутності гілки default, проте семантичні критерії вичерпності явно не сформульовано. Зокрема, залишається незрозумілим, яка часова складність виконання match-конструкцій та як слід трактувати гілки з додатковими умовами if, які покривають варіант лише частково, та гілки, що об'єднують кілька варіантів. Варто було б навести відповідні правила явно.

3. Порівняння Go з альтернативними платформами у підрозділі 4.6 виконано на якісному рівні. На мій погляд, аргументацію посилив би наскрізний приклад: реалізація однієї й тієї самої мовної конструкції, наприклад enum-типів, поверх Go та поверх JVM або LLVM із зіставленням обсягу необхідної інфраструктури.

4. У підрозділі 4.9.1 залежність запропонованого підходу від еволюції самої мови Go лише констатовано із зауваженням, що ризик пом'якшується строгою політикою сумісності Go. Проте для довгострокової життєздатності діалекту не менш важливим є інший аспект цієї залежності: у роботі не розглянуто стратегію супроводу транспілятора при виході нових версій хост-мови, зокрема порядок дій

у разі, коли нова конструкція офіційної версії мови Go виявиться синтаксично несумісною з уже реалізованим розширенням GoNext. Бажано було б доповнити практичні рекомендації розділу 4 відповідними положеннями щодо версіонування діалекту.

5. У роботі запропоновано рішення репрезентації enum-типів на користь безпечних для збирача сміття Go структур (зокрема, з `interface{}` або структурним представленням з полями для всіх варіантів), що змусило автора відмовитися від найбільш ефективних з погляду просторової складності структур даних (зокрема, compact tagged unions). Вибір такого підходу спричиняє додаткові накладні витрати під час виконання програми: структурне представлення вимагає місця в пам'яті для полів усіх варіантів, навіть якщо в момент часу використовується лише один із них, а представлення через `interface{}` — непрямої адресації та можливих додаткових виділень пам'яті.

6. Задля реалізації специфічного синтаксису автором замість побудови повноцінного незалежного проміжного представлення (IR) обрано концепцію прямої трансформації синтаксичних дерев (AST), що спростило реалізацію, проте призвело до таких проблем, як ускладнення збереження точних координат токенів (`token.Pos`), розбухання AST через вираження високорівневих конструкцій (`enum`, `match`) значною кількістю базових вузлів Go та необхідність поєднувати семантичний аналіз із багатопрохідною перебудовою дерева.

Наведені зауваження мають характер побажань і уточнень, не зачіпають основних наукових положень дисертації та не впливають на загальну високу оцінку виконаного дослідження.

Зміст та результати дисертації відповідають спеціальності 121 Інженерія програмного забезпечення, а структура роботи узгоджується з її назвою, метою і завданнями дослідження.

Загальний висновок. Дисертаційна робота Форкерта Павла Павловича «Дослідження використання рантайму Go як платформи для побудови нових мов програмування» є завершеним самостійним науковим дослідженням, у якому отримано нові науково обґрунтовані результати, що в сукупності розв'язують

актуальну задачу інженерії програмного забезпечення. Робота відповідає вимогам «Порядку присудження ступеня доктора філософії та скасування рішення разової спеціалізованої вченої ради закладу вищої освіти, наукової установи про присудження ступеня доктора філософії», затвердженого постановою Кабінету Міністрів України від 12 січня 2022 р. № 44 (зі змінами), а її оформлення — вимогам наказу Міністерства освіти і науки України № 40 від 12.01.2017 р. Здобувач Форкерт Павло Павлович заслуговує на присудження ступеня доктора філософії в галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення.

Рецензент

доцент кафедри інженерії програмного
забезпечення та інформаційних технологій
Дніпровського національного університету
імені Олеся Гончара,
кандидат фізико-математичних наук, доцент

Лілія БОЖУХА

Підпис канд. фіз.-мат. наук, доцента Божухи Л.М. засвідчую

Проректор з наукової роботи
Дніпровського національного
університету імені Олеся Гончара,
кандидат біологічних наук, професор



Олег МАРЕНКОВ