АНОТАЦІЯ

Дейнега О. А. Оптимізація функціональних мов програмування на основі методів штучного інтелекту. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 122 – Комп'ютерні науки (Галузь знань 12 – Інформаційні технології). – Харківський національний університет імені В. Н. Каразіна, Міністерства освіти і науки України, Харків, 2024.

Дисертація присвячена оптимізації функціональних мов програмування, на основі методів штучного інтелекту, що є складною та важливою задачею з багатьма проблемами та викликами. В дисертації розглянуто лямбда-числення як приклад відносно простої репрезентації функціональних мов програмування, що дозволяє показати процеси компіляції та інтерпретації функціональних мов програмування шляхом редукції лямбда-термів.

У **першому розділі** описано теоретичну частину дослідження. Представлено функціональні мови програмування, як інструмент верифікації програмного забезпечення. Описано переваги функціональних програм, такі як простота тестування та надійність коду, а також їх недоліки, основним з яких є низька продуктивність. Далі постає описання лямбда-числення, як одного з найпростіших варіантів представлення функціональних мов програмування. Пояснюється можливість переходу від роботи з функціональними мовами програмування до лямбда-числення. Далі представлені підходи для оптимізації лямбда-числення, основним із яких є удосконалення стратегій редукції лямбда-термів.

Далі текст заглиблюється в зв'язок між лямбда-численням і верифікацією програм в контексті паралельного програмування. Підкреслюється, як лямбда-числення служить основою для аналізу та розуміння поведінки паралельних програм. Використовуючи властивості лямбда-числення, розробники можуть застосовувати формальні методи перевірки, щоб переконатися, що паралельні

програми виконуються правильно та відповідають призначеним специфікаціям. Це включає перевірку таких властивостей, як безпека, живучість і правильність у різних сценаріях виконання.

Описано важливість формальної перевірки для паралельних програм, особливо з огляду на потенційні складності та проблеми, пов'язані з одночасним виконанням. Використовуючи формальні методи, засновані на лямбда-численні, розробники можуть отримати впевненість у надійності та правильності свого паралельного програмного забезпечення.

Далі у першому розділі розглянуті бібліотеки Python для роботи з лямбда численням, виведені їх недоліки та для усунення виявлених недоліків розроблено власну бібліотеку Lambda Calculus Environment. Показано класову архітектуру розробленої бібліотеки та зазначено функції, що виконують зазначені класи. Також зазначено вимоги яким задовольняє розроблена бібліотека і вимоги до функціонування розробленого програмного забезпечення. Згідного вимог до розробленого програмного забезпечення визначено багатоетапну процедуру верифікації. Процедура визначає певний набір лямбда-термів із ключовими умовами які мають бути виконані й враховані при виконанні на них доступних операції. Також процедура визначає перевірку виконання кожного із зазначених пунктів при додаванні нового функціоналу, та перевизначення множини лямбда-термів для тестування.

У **другому розділі** представлений підхід до оптимізації стратегій редукції, що базується на змішуванні стратегій та використанні рандомізованих стратегій. Ідея даного підходу відноситься до теорії ігор, де в деяких випадках використання стратегій в чистому вигляді не може призвести до перемоги. Описані результати, що показують ефективність даного підходу, та можливість заміни чистих стратегій змішаними, що дозволяють зберегти існуючу продуктивність, проте підвищити загальну вірогідність успішного редукування термів.

Далі у розділі була розглянута концепція обчислювальної нерівнозначності редексів лямбда-термів, що є ключовими точками у виборі стратегії редукції.

Нерівнозначність була оцінена з використанням методів машинного навчання для вирішення задачі регресії. Ціллю регресії була оцінка часу виконання операції редукції для даного редексу по параметрам терму, що відображають його деревну структуру. В результаті було отримано відхилення від очікуваного логарифму часу в 0.28 для регресійної моделі на базі штучної нейронної мережі та в 0.28 для лінійної регресії, варто зазначити також незначне падіння точності на тестовому наборі, що свідчило про достатню спроможність зазначених методів вилучати необхідні характеристики для оцінки часу редукції. Застосування методів дерев рішень та опорних векторів для вирішення цієї задачі також не показали високих результатів точності.

Далі у розділі розглядається використання даних про стан терму до і після процесу редукції для оцінки часу, проте без значних покращень у підвищенні точності оцінки часу редукції. Показано, що подальші дослідження цих аспектів можуть збільшити точність оцінки обчислювальних витрат. Також показано, що точна оцінка обчислювальних витрат може допомогти розробити жадібну стратегію з мінімізацію витраченого часу на процес редукції проте без урахування кількості кроків редукції.

У **третьому розділі** була перевірена можливість оцінки кількості кроків редукції лямбда-термів за заданою стратегією із застосуванням методів глибинного навчання. Для цього було використано спрощене текстове представлення лямбда термів із видаленням інформації про змінні. Аналіз проводився з використанням методів глибинного навчання для аналізу послідовностей. Показано, що точних результатів оцінки можливо досягти при визначенні 0-2 кроків редукції проте із збільшенням очікуваних кроків редукції зростала помилка в оцінці. Це свідчило про втрату важливої інформації при використанні спрощеного представлення лямбда-термів та недостатньої спроможності використаних моделей до глибинногоаналізу термів.

Далі було досліджено можливість використання вбудовувань для репрезентації різниці в редукції лямбда термів різними стратегіями. Для цього було

розглянуто чотири моделі LLM для генерації вбудовувань з текстових представлень лямбда-термів. Це дозволяє проаналізувати можливість використання вбудовувань, отриманих з LLM, для вилучення характеристик, що впливають на редукцію термів та в перспективі можуть допомогти розробити компілятори та інтерпретатори функціональних мов програмування.

Згенеровані вбудовування були використані для створення восьми наборів даних для кожної розглянутої моделі LLM та для стратегій редукції термів LO та RI. Ці набори даних містять вбудовування по обраній LLM та кількість кроків редукції для кожного з розглянутих термів за обраною стратегією редукції. Для оцінки якості репрезентації інформації у вбудовуваннях були використані моделі ШНМ, що вирішували проблему класифікації відносно кроків редукції від 0 до 30.

Далі навчені моделі ШНМ були оцінені з показниками для оцінки точності регресії MAE, RMSE. Ці коефіцієнти було порівняно з найкращими результатами, досягнутими для того самого завдання та набору даних зі спрощеним представлення термів. Результати вказують на покращення прогнозування кількості кроків, особливо значних покращень було досягнуто в прогнозуванні кількості кроків LO, що збільшило точність на 23% для показника MAE та на 30% для показника RMSE. Проте прогнози щодо кількості кроків для стратегії RI мали незмінно низький рівень помилок із незначними покращеннями показників MAE та RMSE. Такі результати вказують на те, що код і загальні LLM можуть допомогти отримати інформацію з лямбда-термів і використовувати цю інформацію для вибору оптимальної стратегії редукції. Спеціально навчені LLM можуть ще більше підвищити точність вилучення даних із лямбда-термів. Отже, метод вилучення ознак із використанням LLM може бути реалізований в реальних інтерпретаторах функціональних мов програмування, а вилучені дані можуть бути використані для оптимізації.

У **четвертому розділі** лямбда-терми були перетворені в усереднені вектори вбудовувань розміром 768, що були отримані в результаті застосування попередньонавченої моделі ШНМ для задач пов’язаних із аналізом програмного

коду Microsoft CodeBERT та подальшої обробки виходів середніх рівнів цієї моделі за принципом об’єднання слів у значущі вектори Word2Vec. Завдяки аналізу PCA та t-SNE візуалізацій цих усереднених векторних вбудовувань виявлено, що представлення лямбда-термів у цих усереднених вбудовуваннях можливо візуально чітко розрізнити. Це дозволило підтвердити гіпотезу можливості виділення значущих кластерів в цьому наборі вбудовувань. Також виходи CodeBERT моделі були оброблені із застосуванням автокодувальника проте це не дало бажаної точності візуального розділення даних.

Тому далі було досліджене формування кластерів даних із застосуванням методу DBSCAN, що використовує як евклідову, так і косинусну метрику, окрім методу агломеративної кластеризації з використанням евклідової, косинусної, L1 і L2 метрики. Ці зусилля з кластеризації підкреслили ефективність моделі CodeBERT у вилученні значущих характеристик із лямбда-термів. Незважаючи на це, універсальність Microsoft CodeBERT, навченого різними мовами програмування, вносить рівень складності в досягнення точного представлення термів лямбда-числення в матрицях вбудовувань. Ця складність поширюється на процес перетворення цих матриць у зрозумілі усереднені вбудовування або вектори прихованого простору, особливо при використанні автокодувальників.

Далі була оцінена інформативність змінних усереднених вбудовувань із застосуванням моделі ШНМ навченої на результатах кластеризації та градієнтного методу оцінки інформативності моделі ШНМ. Цей аналіз дозволяє краще зрозуміти вплив певних змінних на результати кластеризації, пропонуючи пояснення основного значення цих змінних у контексті лямбда-термів і мінливості результатів кластеризації.

Крім того, далі було запроваджено коефіцієнт перекриття, що полегшило оцінку взаємозалежності між кластерами та застосованими стратегіями. Ця оцінка виявила відсутність кореляції між попередньо визначеними пріоритетами стратегії та фактично досягнутою дискримінацією термів, що вказує на потенційну потребу

в тонкому налаштуванні моделі CodeBERT та вказує на необхідність розгляду альтернативних моделей, більш придатних для аналізу даних у цій області.

Також було продовжено ідею трансформації лямбда-термів у вектори вбудовувань з використанням моделей OpenAI з розміром векторів 1536, та 3072. Дані вектори були так само проаналізовані з застосуванням методів PCA та t-SNE для візуалізації цих векторів. Так само було виявлено можливість візуального розділення цих даних. Далі також було розглянуто формування кластерів даних із застосуванням методу DBSCAN з евклідовою метрикою. Також підкреслено здатність моделей OpenAI Embeddings вилучати важливі характеристики з лямбда-термів. Однак, навчання OpenAI Embeddings моделей проводилось не на представленнях лямбда-термів, а здебільшого на людському тексті та коді, що ускладнило точне зображення термів лямбда-числення в матрицях вбудовування.

В наступній частині четвертого розділу представлено підхід длявикористання LLM безпосередньо для проведення процесу редукції лямбда-термів. В даному підході на вхід моделі подаються лямбда-терм, а на виході очікується наступний крок редукції лямбда терму за обраною стратегію. Результати показали, що використання LLM для вирішення цієї задачі не є достатньо ефективним.

В останньому розділі дисертації представлено можливий варіант імплементації описаних методів для використання у компіляторах для підвищення їх продуктивності. Описано можливі ризики, що мають бути враховані при використанні даного підходу. Також у даній частині представлено варіант використання великих мовних моделей у якості засобу верифікації лямбда-термів, та функціональних програм в цілому. Оскільки великі мовні моделі мають великий потенціал з точки зору аналізу коду та можуть бути розвинені для забезпечення його надійності.

Сукупність результатів, викладених у дисертації, разом із підтвердженою науковою та практичною актуальністю демонструють досягнення поставленої мети щодо оптимізації функціональних мов програмування на базі методів штучного інтелекту. Цей успіх пояснюється впровадженням сучасних моделей і методів

машинного навчання та штучного інтелекту, що показали високі результати у вирішенні задач даного класу. Крім того, наукові та практичні результати, представлені в дисертації, у поєднанні з перевіркою їх достовірності та значущості, демонструють, що проблема оптимізації функціональних програм за допомогою методів штучного інтелекту була ефективно вирішена, і поставлена мета була досягнута.

Ключові слова: *штучний інтелект, великі мовні моделі, кластеризація, глибинне навчання, інформативність характеристик, машинне навчання, нейронні мережі, профілювання процесу редукції, графове представлення, моделювання лямбда-числення, функціональні мови програмування, автоматизація виборустратегії редукції, симуляція процесу редукції, верифікація програмного забезпечення.*