

Л.І. Мещеряков¹, М.М. Одновол¹, М. Потемпа², В.О. Рибка¹

¹Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

²Технічний університет «Краківська гірничо-металургійна академія ім. Станіслава Сташиця» (AGH), Краків, Польща

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ РОЗПІЗНАВАННЯ НЕІСТИННОСТІ ВИСЛОВЛЮВАНЬ

Анотація. Описано процес пошуку та застосування можливих шляхів оптимізації роботи програмної реалізації методу Куайна – Мак-Класкі на мові програмування C#. Проаналізовано доцільність використання знайдених шляхів оптимізації, а також їх ефективність.

Ключові слова: істинність, розпізнавання істинності міркувань, обчислювальна система, штучний інтелект, числення висловлень, метод Куайна – Мак-Класкі.

Вступ. На даному етапі розвитку інформаційних технологій, штучний інтелект має один з найвищих пріоритетів у розвитку. Саме він вважається майбутнім для усього всесвіту і тому багато уваги та досліджень присвячені саме найрізноманітнішим технологіям та інноваціям, спрямованим на розвиток даної сфери. В основі своєї роботи штучний інтелект має методи та алгоритми алгебри логіки. Таким чином, навіть найменші наукові дослідження в цій сфері можуть привести до швидшого розвитку штучного інтелекту як технології майбутнього. І серед багатьох різноманітних за призначенням методів обчислення висловлень існують методи перевірки виводимості логічних формул, які відіграють важливу роль при реалізації систем штучного інтелекту. Одним з таких методів є метод Куайна – Мак-Класкі. Представлене в роботі дослідження спрямоване на виявлення можливих шляхів оптимізації цього метода при його програмній реалізації на мові програмування C#.

Постановка задачі. Для досягнення поставленої мети в роботі сформульовані і вирішені такі завдання:

- викласти основні поняття алгебри логіки та числення висловлень;
- визначити основні методи перевірки виводимості формул висловлювань;
- визначити основні проблеми, що виникають в програмній реалізації методу Куайна – Мак-Класкі;
- знайти можливі шляхи оптимізації програмної реалізації методу Куайна – Мак-Класкі;
- проаналізувати доцільність застосування методів оптимізації, а також результати їх застосування.

Основний зміст роботи.

Під час роботи були використані наступні методи та інструменти.

- методи алгебри логіки;

- методи числення висловлень;
- методи перевірки виводимості логічних формул;
- C#, Garbage Collector, ObjectPool.

На даному етапі розвитку інформаційних технологій штучний інтелект став одним з передових напрямків розвитку. Штучний інтелект (ШІ) — здатність інженерної системи обробляти, застосовувати та вдосконалювати здобуті знання та вміння. Існують різні методи створення систем штучного інтелекту. Серед них виділяють такі 4 основні.

1. Логічний підхід. Основою для вивчення логічного підходу слугує алгебра логіки. Своєю подальшою розвинутою алгебра логіки отримала у вигляді числення предикатів – в якому вона розширена за рахунок введення предметних символів, відношень між ними.

2. Структурний підхід. Спроби побудови систем штучного інтелекту шляхом моделювання структури людського мозку. Базується на перцептронах. Головною моделюючою структурною одиницею в перцептронах є нейрон.

3. Еволюційний підхід. Під час побудови системи штучного інтелекту за даним методом основну увагу зосереджують на побудові початкової моделі і правилах, за якими вона може змінюватися (еволюціонувати).

4. Імітаційний підхід. Цей підхід є класичним для кібернетики з одним із її базових понять чорного ящика. Об'єкт, поведінка якого імітується, якраз і являє собою «чорний ящик». Для не важливо, які моделі у нього в середині і як він діє, головне, щоб модель в аналогічних ситуаціях поводи́ла себе так же.

Основна увага в роботі зосереджена на логічному підході, зокрема на численні висловлень, його особливостях, а також методах, за якими можна перевірити виводимість інформації.

Числення висловлень є формальною системою, в якій формули, що відповідають висловленням, можуть утворюватися шляхом з'єднання простих висловлень із допомогою логічних операцій, та система правил виводу, які дозволяють визначати певні формули як «теореми» формальної системи.

Серед методів та алгоритмів перевірки виводимості формул можна виділити наступні:

- Алгоритм істинних таблиць (АІТ);
- Метод резолюцій;
- Алгоритм Куайна, який був згодом покращений Едвардом Мак-Класкі.

Для подальших досліджень взято до розгляду алгоритм Куайна – Мак-Класкі. Він є формалізованим на етапі знаходження простих імплікант методом Куайна.

Алгоритм дії даного методу наступний:

1. Терми (кон'юнктивні у разі ДДНФ та диз'юнктивні у разі ДКНФ), на яких визначено функцію алгебри логіки (ФАЛ) записуються у вигляді їх двійкових еквівалентів;

2. Ці еквіваленти розбиваються на групи, до кожної групи входять еквіваленти з рівною кількістю одиниць (нулів);

3. Проводиться попарне порівняння еквівалентів (термів) у сусідніх групах з метою формування термів нижчих рангів;

4. Складається таблиця, заголовком рядків у якій є вихідні терми, а заголовком стовпців – терми низьких рангів;

5. Розставляються мітки, що відбивають поглинання термів вищих рангів (вихідних термів) і далі мінімізація проводиться у спосіб Куайна.

Специфіка методу Куайна – Мак-Класкі в порівнянні з методом Куайна в скороченні кількості попарних порівнянь щодо їх склеювання. Скорочення досягається за рахунок вихідного розбиття термів на групи з рівною кількістю одиниць (нулів). Це дозволяє виключити порівняння, що свідомо не дають склеювання.

В програмній реалізації даного методу було виявлено наступні недоліки його роботи:

- швидкість роботи алгоритму. Як відомо швидкість роботи даного методу і справді є його найбільшим недоліком, тому отримання цього висновку є очікуваним;

- при великій кількості вхідних змінних в термах робота програми завершується через нестачу пам'яті.

Оскільки швидкість роботи алгоритму є його «вродженою» вадою, можна зосередити увагу на вирішенні проблем, пов'язаних з нестачею пам'яті при роботі алгоритму. Слід зазначити, що проблема нестачі пам'яті виникає тільки в тому випадку, якщо оптимізація відбувається на наборі вхідних терм розміром трохи менше максимальної кількості 2^N , де N – кількість змінних. Якщо ж використовується скорочений (неповний) набір з кількістю $Q < Q_{\text{пор}} < 2^N$, то проблема не виникає доти, поки розмір вхідного набору даних не переросте деякий поріг $Q_{\text{пор}}$, критичний для алгоритму.

В якості вирішення виявлених проблем пропонуються наступні рішення.

1. Визначення вузлів троїчного дерева як структур.

2. Додавання до програми пулу об'єктів.

3. Явний виклик методів Garbage Collector.

Результатом аналізу запропонованих підходів для оптимізації роботи даного методу є наступне:

– Ідея застосування структур є неможливою для даного методу через свою вбудовану властивість – неможливість мати в собі значення NULL, при тому що деякі елементи реалізації можуть потенційно містити в собі таке значення.

– Застосування пулу об'єктів є недоцільним для даного методу. Головна ідея пулу об'єктів полягає в повторному використанні об'єктів програми, та при цьому необхідно повертати їх назад до пулу. Проте, проблема нестачі пам'яті в представляємій реалізації методу Куайна – Мак-Класкі відбувається на етапі масового створення об'єктів, які будуть використовуватись надалі, і виходячи з цього, ще не можливо повернути їх до пулу об'єктів для повторного використання.

– Явний виклик методів збирача сміття є застосовним і ніщо не заважає його використовувати.

Після аналізу можливості застосування кожного з варіантів покращення роботи методу Куайна – Мак-Класкі практично реалізовано та протестувано роботу програму з застосуванням явного виклику методів збирача сміття. Суть роботи збирача сміття полягає в наступному. Під час процесу збирання сміття виконуюче середовище досліджуватиме об'єкти в купі, щоб визначити, чи є вони, як і раніше, досяжними для програми. Для цього середовище CLR буде створювати графи об'єктів, що представляють всі доступні для програми об'єкти. Крім того, слід мати на увазі, що збирач сміття ніколи не створюватиме граф для одного і того ж об'єкта двічі, позбавляючи необхідності виконання підрахунку циклічних посилань, який характерний для програмування в середовищі COM. Щоб змусити збирача сміття провести прибирання, достатньо викликати метод `Collect`, доступний у класі `GC`.

Тестування отриманої програми відбувалось на двох різних машинах з різними технічними показниками. В результаті отримано висновок про те, що ідея застосування збирача сміття дійсно допомагає вирішити проблему з нестачею пам'яті, і є два можливі варіанти, чому:

– збирач сміття таки збирає на кожній ітерації деякі об'єкти, тим самим звільняючи пам'ять;

– `GC.Collect()` дефрагментує пам'ять, завдяки чому туди може «влізти» більше об'єктів типу `TreeNodeEnd`.

Проте, в ході тестування було також виявлено ще один важливий результат. При застосуванні методів збирача сміття в деяких випадках була збільшена швидкість роботи програми, хоча очікувалось зворотне – зменшення швидкості роботи за рахунок витрат часу на виклик методу збирання сміття, а також його виконання. Особливо помітним цей ефект виявився при тестуванні програми на комп'ютері з більш низькими технологічними показниками, а також таких, які мають більш «засмічену» пам'ять. Цей ефект можна пояснити тим, що збирач сміття зменшує дефрагментацію пам'яті, що в свою чергу зменшує час на пошук вільних ділянок пам'яті для нових об'єктів.

Наукова новизна полягає в наступному: дослідження можливих способів оптимізації роботи алгоритму Куайна – Мак-Класкі на платформі .NET; реалізація одного з варіантів оптимізації роботи методу Куайна – Мак-Класкі на мові C#.

Висновки. В результаті виконання роботи було знайдено декілька способів оптимізації роботи методу Куайна – Мак-Класкі на мові програмування C#. Проведено аналіз доцільності використання кожного з запропонованих варіантів. Реалізований варіант оптимізації, що базується на явному виклику методів збирача сміття (`Garbage Collector`). В результаті отримано вирішення не лише проблеми нестачі пам'яті при реалізації методу Куайна – Мак-Класкі на мові C#, а також і покращення в цілому швидкості роботи програми, що його реалізує.

ПЕРЕЛІК ПОСИЛАНЬ

1. Міркес Е. М., Нейрокомп'ютер. Проект стандарту.- Новосибірськ: Наука, Сибірська видавнича фірма РАН, 1999 .- 337 с. ISBN 5-02-031409-9 (Глава 9: «Контрастер»)
2. McCulloch, Warren S.; Pitts, Walter (December 1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* (en) **5** (4): 115–133.
3. Обчислення висловлень. Перевірка виведення правильних висновків. Алгоритм Куайна. Правило резолюцій. [<https://studfile.net/preview/413969/>]
4. Метод Куайна — Мак-Класкі [https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%9A%D1%83%D0%B0%D0%B9%D0%BD%D0%B0_%E2%80%94%D0%9C%D0%B0%D0%BA-%D0%9A%D0%BB%D0%B0%D1%81%D0%BA%D1%96]
5. Fundamentals of garbage collection [<https://docs.microsoft.com/en-us/dotnet/standard/garbage-collection/fundamentals>]
6. Збирання сміття, управління пам'яттю та вказівники [<https://metanit.com/sharp/tutorial/8.1.php>]
7. Garbage Collection in C# | .NET Framework [<https://www.geeksforgeeks.org/garbage-collection-in-c-sharp-dot-net-framework/>]

УДК 004.896

S.I. Kostyrska¹, I.G. Hulina¹, K. Pałasz², I.V. Taran¹

¹Dnipro University of Technology, Dnipro Ukraine

²Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Kraków, Polska

PROSPECTS FOR USING BIPEDAL ROBOTS

Анотація. Двоногі роботи розробляються вченими для покращення життєвих умов. Приклади відносно досягнень у цій галузі пояснюють, як саме подібні роботи можуть стати в нагоді людству.

Abstract. Bipedal robots are developed by scientists to improve life conditions. The examples illustrate how bipedal robots can benefit mankind.

Key words: *bipedal robot, locomotion, “spring - mass”, balance, ATRIAS, Cassie, Mercury, LEONARDO, flying and walking locomotion.*

Introduction. Nowadays, artificial intelligence devices are widely used. Since working in the extreme environment could cause huge damage to a person`s health, the researchers invented the applications to substitute people in hazardous areas, especially when it comes to the fire rescue operations, toxic gases or chemicals.

As a rule, people take walking process for granted. However, in reality, it is not that simple. Before we take a step, the nerves send signal to the brain to define the essential parameters which enable the brain to do the number of calculations, so the