

УДК 004.93

Марченко О.І., к.т.н., доцент,

Лиман Д.М., студент,

Національний технічний університет України «Київський політехнічний інститут»

МОДИФІКОВАНИЙ СПОСІБ ПОВНОГО ПЕРЕБОРУ ДЛЯ ВИЗНАЧЕННЯ ВІДПОВІДНОСТІ ДЕРЕВ

Марченко О.І., Лиман Д.М. Модифікований спосіб повного перебору для визначення відповідності дерев. У статті запропоновано спосіб, який надає можливість більш швидкого визначення повної або часткової відповідності двох дерев, ніж способом повного перебору. Запропонований спосіб є модифікацією способу повного перебору з використанням додаткових структур даних для визначення можливості дострокового завершення чергової ітерації порівняння у випадку знаходження невідповідності між деревами. Результатом роботи способу є множини верши та ребер з частковою або повною відповідністю між деревами, що порівнюються.

Ключові слова: граф, дерево, ізоморфність, відповідність дерев, NP задача.

Марченко А.И., Лиман Д.Н. Модифицированный способ полного перебора для определения соответствия деревьев. В статье предложен способ, который предоставляет возможность более быстрого определения полного или частичного соответствия двух деревьев, чем способ полного перебора. Предложенный способ является модификацией способа полного перебора, с использованием дополнительных структур данных для определения возможности досрочного завершения очередной итерации сравнения в случае обнаружения несоответствия между деревьями. Результатом работы способа являются множества вершин и ребер с частичным или полным соответствием между сравниваемыми деревьями.

Ключевые слова: граф, дерево, изоморфность, соответствие дерев, NP задача.

Marchenko O.I., Lyman D.M. Modified brute force algorithm for determining trees correspondence. The proposed method allows quicker determining the complete or partial correspondence between two trees compared to brute force method. Presented method is a modification of brute force method. It uses additional structures to determine the possibilities of early termination of the current comparison iteration when discrepancy between trees detected. The method produces sets of vertices and edges with partial or full trees correspondence.

Keywords: graph, tree, isomorphic, tree congruence, NP task.

Рис. 1, Лим. 9.

Постановка наукової проблеми.

Задача визначення відповідності дерев є частковим випадком більш загальної задачі визначення ізоморфності графів [1], яка використовується у багатьох сферах де можна спростити реальну модель до дерева. Наприклад, одним із застосувань у сфері автоматизації проектування електронних схем є верифікація представлення різних схем, що виконується завдяки пошуку відповідності між графами, при статичному аналізі вихідного коду деякі способи знаходження дублікатів побудовані на основі пошуку подібності між деревами. Загальний випадок рішення задачі ізоморфізму є задачею, яку відносять до групи NP-повних задач [2]. Частковий випадок цієї задачі, коли визначається відповідність між деревами, може бути спрощений завдяки відсутності будь-якого виду циклічності у деревах, наявності однієї кореневої вершини, від якої будуються зв'язки з іншими вершинами, та іншими характеристиками дерева згідно теорії графів. Пошук способів прискорення процесу визначення ізоморфності дерев на основі їх більш простої структури, ніж у графів, є актуальною задачею.

Аналіз досліджень.

Задача визначення ізоморфності двох графів, яка є більш загальною задачею розглянутої у цій статті, є NP-повною. Вирішенню NP-повних задач присвячена велика кількість наукових статей, а в деяких з них намагаються вирішити часткові випадки [3]. Як відомо, дерева є підмножиною ненаправлених графів зі своїми властивостями, що спрощують можливість визначення відповідності.

У статті [4] пропонується вирішення проблеми визначення входження дерев завдяки створенню структури даних, яка задає множину процедур для виконання алгоритму повного перебору. Класичним способом вирішення даної задачі вважається алгоритм Ульмана [5]. Способи, що мають на меті скорочення кількості дій для встановлення відповідності відрізняються між собою підходами для вирішення цієї задачі. Наприклад, існують способи, що аналізують вхідні моделі та застосовують структуровані методи прогнозування результату [6] або використовують трансформації для зведення проблеми до задачі пошуку максимальної кліки у графах та побудови допоміжних структур даних у

вигляді графів асоціацій, використовуючи концепцію зв'язності теорії графів [7].

Метою даної роботи є розроблення способу для прискорення процесу визначення відповідності між вершинами двох вхідних дерев завдяки частковому обходу та побудові словника повної або часткової узгодженості вершин.

Повний перебір вхідних дерев для визначення відповідності дерев.

Проблема визначення відповідності дерев зводиться до визначення множин вершин, які необхідно видалити, щоб отримати дерево T з дерева P . Наведемо деякі визначення та властивості дерев, які будуть використані надалі у статті:

- усі вершини дерева мають зв'язки;
- існує коренева вершина, від якої можна знайти простий унікальний шлях до будь-якої вершини;
- зв'язки типу «батько-дитина» та «дитина-батько» – зв'язки між вершинами, що з'єднані напряму одна з одною, рухаючись від кореня та в зворотньому напрямку відповідно;
- вершини з однаковою довжиною шляху від кореня, утворюють ярус дерева.

З властивостей дерева витікає відсутність гамільтонових циклів, які наявні у графах, що суттєво зменшує кількість можливих дій для отримання одного дерева з іншого. У той же час, поняття впорядкованості дерев ускладнює пошук відповідності. Граф називається впорядкованим, якщо існує і враховується порядок між вершинами-дітьми. Вирішення задачі відповідності невпорядкованих дерев є більш складним випадком, оскільки виникає проблема визначення гомоморфних дерев. Це означає, що для визначення відповідності між структурами двох дерев необхідно брати до уваги порядок між дітьми одного рівня. На рисунку 1 наведено два невпорядковані дерева, які мають відповідність.

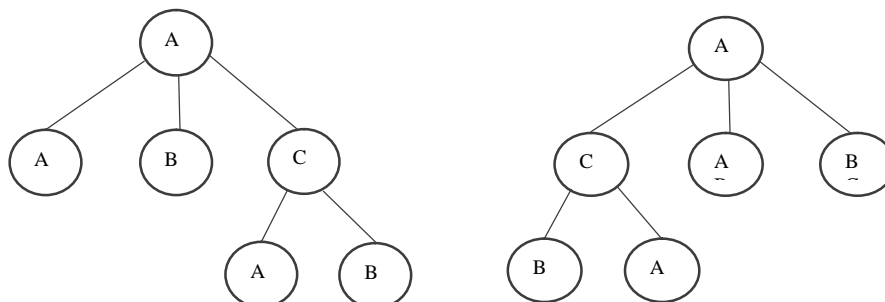


Рис. 1. Невпорядковані гомоморфні дерева

Повний перебір є загальним способом, який полягає у систематичному переборі усіх можливих варіантів для знаходження рішення поставленої задачі за допомогою послідовної перевірки кожного кандидата на відповідність критерію рішення цієї задачі. Хоча спосіб повного перебору легкий у реалізації і в результаті буде точно знайдено рішення, якщо воно існує, але при його використанні можливі випадки комбінаторного вибуху, тобто різкого зростання часу, потрібного для визначення відповідності дерев, через збільшення кількості можливих рішень, які необхідно перевірити. Цей спосіб може бути достатньо ефективно використаний у разі, якщо вдається знайти шляхи зменшення множини рішень, яку потрібно аналізувати, до прийнятної кількості елементів. У випадку, якщо простота або наочність вирішення задачі має вищий пріоритет за швидкість вирішення самої задачі, використання способу повного перебору цілком виправдане.

Спосіб повного перебору обов'язково повинен складатися з чотирьох етапів: ініціалізація можливого варіанту, генерація наступного, валідація та отримання вихідного результату. Перелічені процедури приймають в якості вхідного параметру P дані, що відображають проблему, яку намагаються вирішити. Порядок дій та їх опис наступний:

1. Ініціалізація (P): генерація першого кандидата для вирішення поставленої проблеми P .
2. Наступний (P, c): генерація іншого кандидата c для P .
3. Валідація (P, c): перевірка чи задовольняє кандидат c проблему P .
4. Вихідні дані (P, c): використання кандидата c для проблеми P як придатний результат.

Додатковою задачею другої процедури є показ випадку відсутності інших кандидатів рішення проблеми P , окрім кандидата, згенерованого в результаті першої процедури. Загальний підхід до

такого випадку є повернення так званого null-кандидата (нульового кандидата), тобто такого значення кандидата, яке не входить в множину можливих значень реальних кандидатів для вирішення проблеми P . Також можлива ситуація, при якій взагалі відсутні кандидати для вирішення поставленої задачі P , тому повернення null-кандидата як результат роботи є валідним результатом.

Спосіб повного перебору може бути представлений у вигляді алгоритму верхнього рівня:

```

c ← ініціалізація(P)
while c не дорівнює null кандидату do
  if валідація(P,c) then
    вихідні дані(P, c)
  end if
  c ← наступний(P,c)
end while
    
```

В загальному випадку, спосіб повного перебору може бути легко модифікований для використання першого кандидата, що задовольняє процедурі валідації, в якості кінцевого результату. Також можливе використання інших критеріїв для зупинки пошуку, таких як, наприклад, припинення виконання пошуку після отримання заданої кількості придатних результатів для вирішення заданої проблеми P або після певного часу роботи (тактів процесора) програми, що реалізує даний спосіб, тощо.

Крім того, для реалізації пошуку відповідності між вхідними деревами повним перебором необхідно визначити також критерій перевірки кандидатів на коректність результату та описати дії чотирьох процедур, що були зазначені вище. Кандидатами перевірки коректності є піддерева одного вхідного дерева, які порівнюються для пошуку відповідності з іншим вхідним деревом. На кожній ітерації процесу визначення відповідності дерев відбувається видалення певної вершини з метою отримання піддерева з кількістю вершин рівною кількості вершин того дерева з яким визначається відповідність. Оскільки дерева завжди мають кореневу вершину, то піддерево з кореневою вершиною і буде кандидатом при ініціалізації пошуку, а кожний наступний кандидат буде піддеревом з видаленою однією вершиною з минулого кандидата. Процедура валідації двох дерев – це перевірка на еквівалентність дерев отриманих шляхом всіх можливих перестановок між вершинами дерева кандидата та дерева з яким визначається відповідність. Критерієм еквівалентності неупорядкованих дерев є відповідність зв'язків між вершинами графа. У випадку упорядкованих дерев, до критерію неупорядкованих дерев додається критерій збереження порядку між дітьми одного рівня. У разі позитивного результату на етапі валідації, на виході отримаємо два дерева, які мають вершини з повною відповідністю між собою.

Модифікований спосіб перебору дерев для визначення відповідності неупорядкованих дерев

В запропонованому способі змінюються кандидати для порівняння та критерій верифікації двох дерев. Як зазначено вище, критерієм відповідності двох дерев є однаковість зв'язків між вершинами. Зв'язки між вершинами дерев можна представити у вигляді списку пар ключ-значення, так званого словника, де ключем є вершина-батько, а значенням – вершина-дитина. Таким чином зберігається характеристика еквівалентності дерев та спрощується визначення кінцевого рішення. Ним має бути один або декілька словників відповідності в яких ключами є вершини дерева з яким визначається відповідність, а значеннями – вершини дерева в якому знаходиться відповідність. Тобто представлені словники відповідності є кандидатами для верифікації, процес порівняння яких має швидку реалізацію у багатьох мовах програмування. В свою чергу змінюється й критерій перевірки, ним стає наявність усіх вершин дерева, що порівнюється у якості ключів в словнику відповідності між вершинами двох дерев.

Основою способу, що пропонується, є спуск по ярусам дерев з побудовою усіх можливих варіантів відповідності між деревами, оскільки вхідні структури даних є невідсортованими і необхідно виконати усі перестановки для визначення можливих варіантів рішення. Наступний етап – пошук помилки у побудованих варіантах відповідності та збільшення кількості елементів словника, поки кількість записів у кандидаті не буде дорівнювати кількості вершин дерева, з яким відбувається процес пошуку відповідності. Процес перебору згідно даного способу складається з трьох кроків:

1. Взяття чергового ярусу дерева.
2. Побудова усіх можливих варіантів відповідності між деревами на ярусі.
3. Злиття кожного варіанту з існуючим словником відповідності.

Відмінність даного способу від способу повного перебору полягає у зменшенні кількості можливих варіантів відповідності пошуком невизначеності у словнику після кожної ітерації. Невизначеність у словнику – це двозначність між ключами або значеннями, тобто два ключі відповідають одному значенню, або навпаки два значення відповідають одному ключу. Внаслідок використання критерія невизначеності у словнику, кількість ітерацій зменшується порівняно зі способом повного перебору. Наведений спосіб закінчує свою роботу у двох випадках: пройдено усі вершини або знайдено двозначність словника.

У разі вичерпання ярусів обох вхідних дерев, можна зробити висновок, що було знайдено шукану відповідність між двома структурами. В загальному випадку, може бути більше ніж один словник повної відповідності між вершинами дерев, оскільки повна перестановка дає велику кількість можливих варіантів. Випадок, коли кожен з можливих варіантів продукує колізію в існуючому словнику відповідності, визначає неповну відповідність між двома деревами. В залежності від кількості вершин, що були отримані в результаті визначення гомоморфізму структур даних, можна підрахувати ступінь відповідності.

Наведемо опис послідовних дій запропонованого способу:

1. Ініціалізація двох списків повної та часткової відповідності пустими словниками.
2. Паралельний обхід двох вхідних дерев, починаючи з першого ярусу. Далі будуть циклічно виконуватися наступні дії:

2.1. Виконання перестановки вершин на поточному ярусі другого дерева по кількості вершин з поточного ярусу першого дерева.

2.2. Створення словника відповідності для кожної перестановки з п.2.1., де ключем є вершина з першого дерева поточного ярусу, а значенням – вершина з другого дерева поточного ярусу.

2.3. Кожен словник з п.2.2. поєднується з кожним елементом зі списку відповідності. Словники, в яких при поєднанні не виникла колізія, є новим значенням списку словників відповідності. Словники, в яких при поєднанні виникла колізія між ключами або значеннями, вважаються словниками часткової відповідності і додаються до списку часткової відповідності, після чого вже не приймають участі у подальшому поєднанні.

2.4. Якщо усі словники з п.2.2 продукують невідповідність у результируючих словниках відповідності, то необхідно перенести усі словники зі списку повної відповідності до списку часткової відповідності, і виконання способу закінчується. В інакшому випадку, виконується взяття наступного ярусу з вхідних дерев та перехід до п. 2.1. У разі, якщо наступного ярусу в одному з вхідних дерев немає, виконання способу закінчується.

3. Результатом виконання дій способу є два списки. В залежності від наявності або відсутності словників в списку повної відповідності, можна зробити висновок про еквівалентність між вхідними деревами.

Оскільки представлений спосіб на виході надає не тільки словник з повною відповідністю вершин, а й множину словників з частковою відповідністю, то за допомогою даних словників можна визначити відсоткове значення співпадіння між вхідними деревами двох типів: за кількістю безпомилкових варіантів відповідності між вершинами та повних ярусів дерев. Відсоткове значення може бути знайдене як відношення кількості вершин з безпомилковою відповідністю до кількості вершин дерева, з яким визначалась відповідність.

Наведене вище відношення може бути модифіковане для пошуку відсотку відповідності між деревами за кількістю повних ярусів дерев, що відповідають безпомилково. Для цього необхідно замінити лише підрахунок кількості вершин на підрахунок кількості ярусів. Якщо інформація про показники дерев, що порівнюються, відсутня, то для повної оцінки результату пошуку відповідності необхідно використовувати обидва показники, а саме: середнє значення вершин у ярусі, середня кількість вершин-дітей та інші.

Отримані словники відповідності можуть бути використані у різних сферах, в яких необхідно визначити відповідність між двома моделями, за умови, що моделі можуть бути представлені у вигляді дерева. При проектуванні електронних схем існує необхідність автоматизація процесу перевірки

різних представлень електронних схем. Також існують способи повної автоматизації оптимізації програмного забезпечення за допомогою пошуку загальних абстрактних синтаксичних дерев для зменшення кількості дублювання у коді, а також часткової автоматизації програмного забезпечення у випадку аналізу зв'язків між структурами даних, змінними та іншими логічними частинами програмного забезпечення [8]. Іншим прикладом використання пошуку відповідності між деревами є аналіз шаблонів проектування у вихідному коді програмного забезпечення. Такий підхід активно використовується у випадку аналізу вихідного коду значного розміру для подальшого автоматизованого або ручного рефакторингу [9].

Визначені часткові відповідності можуть бути використані для рішення задач пов'язаних з ізоморфністю дерев та інших видів направлених графів. Прикладами таких задач є пошук найбільших або найменших спільних піддерев та визначення мінімального розширення вершин та ребер для повної відповідності між вхідними направленими графами.

Висновки

У даній статті пропонується модифікація способу повного перебору, який дозволяє визначати повну або часткову відповідність між двома деревами. Представлений спосіб надає можливість дострокового завершення визначення повної або часткової відповідності між вершинами двох неупорядкованих вхідних дерев, використовуючи загальний підхід повної перестановки між вершинами та список словників як додаткових структур даних для пришвидшення процесу верифікації кандидатів для вирішення поставленої задачі з метою дострокової зупинки процесу пошуку. Додатковим результатом роботи способу є визначення множини вершин та ребер, що необхідно видалити для отримання співпадаючих дерев.

У якості напрямку для подальшої модифікації та розвитку розглянутого способу можна запропонувати реалізацію виконання вибору підмножини можливих варіантів повної відповідності для зменшення кількості виконуваних операцій. Крім того, можна виконувати аналіз вихідного результату для представлення оцінки відповідності двох орієнтованих графів, а також розширити використання способу на інші види графів: неорієнтовані, змішані, циклічні.

1. Kelly. A congruence theorem for trees. / Kelly, Paul J. // Pacific J. Math. – 1957. – P. 961–968.
2. Campbell D.M. Tree isomorphism algorithms: Speed vs. Clarity. / D.M. Campbell // Math. Mag. 64, No. 4. – 199. – P. 252-261.
3. Yuko Itokawa, Masanobu Wada, Toshimitsu Ishii and Tomoyuki Uchida. Tree Pattern Matching Algorithm Using a Succinct Data Structure // Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol I, IMECS 2011, March 16-18, 2011, Hong Kong.
4. Philip Bille and Inge Li Gørtz. The Tree Inclusion Problem: In Optimal Space and Faster // Technical Report IT University of Copenhagen, January 2005
5. J.R. Ullman. An Algorithm for Subgraph Isomorphism // Journal of the ACM (JACM) - Volume 23 Issue 1 - Jan. 1976, pp. 31-42.
6. Ranjitha Kumar. Flexible Tree Matching / Ranjitha Kumar, Jerry O. Talton, Salman Ahmad, Tim Roughgarden, Scott R. Klemmer // Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three, p. 2674-2679, 2011.
7. M. Pelillo. Matching hierarchical structures using association graphs / M. Pelillo, K. Siddiqi, S.W. Zucker // IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 21, 1999.
8. Свамі М. Графи, мережі та алгоритми / М. Свамі, К.Тхулаліраман // Світ. — 1984. — 455с.
9. Кериевски Дж.М. Рефакторинг с использованием шаблонов. / Дж.М. Кериевски // Вильямс. — 2008. — 400 с.