

РЕШЕНИЕ ВОПРОСА ИЗОМОРФИЗМА НЕОРИЕНТИРОВАННЫХ ДЕРЕВЬЕВ МЕТОДОМ ГЕНЕРАЦИИ ИЗОМОРФНЫХ СТРУКТУР

Ключевые слова: граф, дерево, неориентированное дерево, изоморфизм.

Введение

Теория графов [1–6 и др.], являясь одним из обширнейших разделов дискретной математики и выступая в роли универсального средства формализованного описания различного рода процессов, структур и объектов, продолжает привлекать исследователей, а ее практическое использование в целях получения ответов на всевозможные вопросы, возникающие при решении технических, экономических и управленческих задач в программировании, лингвистике, психологии, социологии и других сферах деятельности, по-прежнему сохраняет актуальность. Один из таких вопросов, породив общую по формулировке и, в значительной степени равноценную по значимости задачу, постоянно фигурирует в ряде предметных областей знаний. Цель решения такой задачи — улучшение существующих методов и средств оценки изоморфности структур графов-деревьев, а также поиск и разработка их новых вариантов, наиболее простых и эффективных в использовании как с точки зрения общения с ними пользователей и наглядности представления получаемых финальных результатов, так и со стороны общих затрат ресурсов.

Главная особенность существующих подходов и методов решения подобных вопросов — ориентация на использование в качестве базового материала данных различного рода средств формализованного представления деревьев (СФПД), таких как списки смежности, матрицы инцидентности, смежности, связности и т.д., работа с которыми, как правило, чревата значительными затратами времени и технических ресурсов используемых вычислительных средств, существенно возрастающими при росте числа вершин N исследуемых объектов.

Отмеченное выше послужило причиной разработки нового подхода, позволившего подойти к решению вопроса изоморфизма деревьев с несколько иной стороны. При этом вместо традиционной последовательности этапов решения задачи, а именно, СФПД → (стандартизированные языки описания структуры) → (программы для построения графов) → (средства визуализации графов) → (средства работы с графами), предложен метод условно «обратного» пути ее решения (не путать с методом решения обратной задачи).

В отличие от отмеченного выше, в нашем случае весь процесс работы с графами-деревьями происходит в специализированной визуальной среде созданного комплекса программных средств. Эта среда интегрирует в себе возможности одновременной визуализации, построения, различного рода модификаций структуры деревьев и получения ряда характеристик, тем самым, исключая необходимость освоения и непосредственного использования различного рода специализированных языков, программ и средств. Общение пользователя с комплексом происходит на визуально-интуитивном уровне понимания им сути подлежащих реализации действий и чем-то напоминает игру «LEGO» или «Plug-and-Play» режим работы ПК. При этом СФПД, с которых обычно начиналось решение задачи, теряют статус первоосновы по причине собственной ненужности.

Что же касается конкретно матрицы смежности, то два различных варианта ее представления, характеризующие финальный результат выполнения различного рода операций, могут быть получены и записаны в файлы с расширениями FileName.cam (условно, «classic adjacency matrix») и FileName.tam («truncated adjacency matrix») в конце работы программы и только при наличии у пользователя такого желания. Под усеченным (truncated) вариантом матрицы смежности размерности $N \times N$ понимается такой ее вид, когда используются только данные, расположенные выше ее главной диагонали и общее число которых оценивается величиной $\frac{N(N-1)}{2}$.

Цель публикации — разработка метода (механизмов, способов, алгоритмов и т.д.) и создание на его основе программных средств, решающих задачу анализа изоморфности деревьев без использования существующих средств их формализованного представления, более оперативно, с меньшими затратами вычислительных ресурсов и времени.

Реализация метода

Прежде чем приступить к описанию разработанного метода, приведем ряд определений, используемых при последующем изложении материала, а именно:

изоморфная реконфигурация (перестройка) — операция над изначально заданной структурой любого N -вершинного ($2 < N$) дерева, преобразующая последнюю в аналог, удовлетворяющий всем требованиям изоморфности исходному варианту;

эталонное дерево G^0 — изначально заданное N^0 -вершинное дерево, структура которого остается неизменной на всем промежутке времени решения задачи;

контрольное дерево G^1 — N^1 -вершинное дерево, структура которого подлежит анализу на изоморфность эталонному дереву G^0 и служит основой для построения собственных изоморфных аналогов в процессе решения задачи;

тестируемое дерево G^* — используемое в процессе решения задачи N^1 -вершинное дерево, структура которого в процессе решения задачи формируется из контрольного дерева G^1 в результате применения к нему процедуры изоморфной реконфигурации.

Трудоёмкость решения вопроса изоморфности структур пары деревьев $G^0 = (A^0, B^0)$ и $G^1 = (A^1, B^1)$ (A^* — множество вершин, B^* — множество ребер), существенно возрастает при росте общего числа вершин, когда в роли корней G^0 и G^1 выступают вершины, которые не являются висячими (листьями) или имеют различную степень и т.д. Если же указанные особенности интегрируются воедино, то использование для решения вопроса существующих различного рода СФПД, приводит к многократно возрастающим и еще более значительным ресурсным затратам.

В основе программной реализации разработанного метода решения задачи лежит операция изоморфной реконфигурации структур деревьев, которая после выбора для работы соответствующей структуры (имени файла) и вершины, выступающей в роли нового корня, активизируется отдельной кнопкой приведенного на рис. 1 общего вида интерфейса пользователя и имеет название «7. Изоморфная реконфигурация структуры графа при замене корневой («ROOT») вершины». Реализующая этот метод процедура написана на языке MS VISUAL C++ 6, имеет универсальный характер и не зависит от того, является ли выбираемая в качестве нового корня вершина листом или промежуточной в каком-либо из путей.

РАБОТА С ГРАФАМИ-ДЕРЕВЬЯМИ

ТИПЫ ОПЕРАЦИЙ, РЕАЛИЗУЕМЫХ НА ГРАФАХ-ДЕРЕВЬЯХ (создание, модификация и реконфигурация структуры, изменение данных вершин, получение характеристик...)

Базовые операции на графах-деревьях:

- Создание НОВОЙ структуры графа
- Операции на существующей структуре графа-деревья

Имена файлов РАБОЧЕЙ структуры: 1_Tree.trt

Число вершин в РАБОЧЕЙ структуре: 21

Вывод информации по выполненной операции

Имена файлов сортированных структур: 1_Tree.trt

Число вершин в выбранной структуре: 21

Отменить выполнение выбранной операции

ОПЕРАЦИИ МОДИФИКАЦИИ СТРУКТУРЫ РАБОЧЕГО ВАРИАНТА ГРАФА

Добавление в структуру графа вершин на указанном месте

0. На АКТИВНОМ X-уровне 1. На 1 выше АКТИВНОГО X-уровня 2. На 1 ниже АКТИВНОГО X-уровня

Удаление компонентов структуры графа (вершины, вершины и порожденной ею ветки)

3. Вырезать маркированную вершину (ветку)

Вырезать маркированную вершину

Вырезать маркированную вершину с веткой

Копирование компонентов структуры графа (вершины, вершины и порожденной ею ветки)

4. Копировать маркированную вершину (ветку)

Копировать маркированную вершину

Копировать маркированную вершину с веткой

Место ветки корня выраженного (скопированного) компонента в структуре графа-деревья

5. На один Y-уровень выше АКТИВНОГО 6. На один Y-уровень ниже АКТИВНОГО

На X-уровне аналогично АКТИВНОМУ

На один X-уровень ниже АКТИВНОГО

Изолированная реконфигурация структуры графа при замене корневой (ROOT) вершины

7. Реконфигурировать исходящую структуру графа для НОВОЙ ROOT вершины

Ветка за веткой Вершина за вершиной

8. Маркировать вершину 9. Показать путь от вершины к "корню"

Данные выбранной вершины: 0 Комментарий к данной вершине

10. Перенумеровать вершины 11. Найти вершину по "личному" номеру

12. Указать левую для анализа изоморфности структур графа (для пары графов-деревьев)

13. Получить левую для анализа изоморфности структур графа (для пары графов-деревьев)

Показать результат выполнения операции на дереве

ОКНО ВЫВОДА СПРАВОЧНОЙ ИНФОРМАЦИИ ОБЩЕГО ТИПА А ТАКЖЕ ПО КАЖДОЙ ПРЕДПОСМОТЕННОЙ ПРОГРАММНОЙ ОПЕРАЦИИ

14. ***** ИНФОРМАЦИЯ ОБЩЕГО ХАРАКТЕРА *****
 Появляется в специализированном информационном окне в момент начала работы программы. Служит для предоставления пользователю возмож-
 ности ознакомиться с основными характеристиками разработанной и протестированной в настоящем программном продукте базовой функ-
 ции. Для удобства вверху в левом углу экрана выделены основные термины, которые будут использоваться в дальнейшем. Внизу экрана выделены
 для удобства вверху в левом углу экрана выделены основные термины, которые будут использоваться в дальнейшем. Внизу экрана выделены

Выборить НОВЫЙ режим работы с графами-деревьями

Завершить работу программы

Структура графа-дерева (2_Тее) _гестн_ _гестн_
Идеальная на изоморфность "углового"

```

ROOT № 0; X_Level = 0; Mark = 4; (ADD Param = 0);
  1 Node № 1; X_Level = 1; Mark = 1; (ADD Param = 1);
  2 Node № 2; X_Level = 1; Mark = 1; (ADD Param = 1);
  3 Node № 3; X_Level = 1; Mark = 1; (ADD Param = 1);
  4 Node № 4; X_Level = 1; Mark = 1; (ADD Param = 1);
  5 Node № 5; X_Level = 1; Mark = 1; (ADD Param = 1);
  6 Node № 6; X_Level = 1; Mark = 1; (ADD Param = 1);
  7 Node № 7; X_Level = 1; Mark = 1; (ADD Param = 1);
  8 Node № 8; X_Level = 1; Mark = 1; (ADD Param = 1);
  9 Node № 9; X_Level = 1; Mark = 1; (ADD Param = 1);
  10 Node № 10; X_Level = 1; Mark = 1; (ADD Param = 1);
  11 Node № 11; X_Level = 1; Mark = 1; (ADD Param = 1);
  12 Node № 12; X_Level = 1; Mark = 1; (ADD Param = 1);
  13 Node № 13; X_Level = 1; Mark = 1; (ADD Param = 1);
  14 Node № 14; X_Level = 1; Mark = 1; (ADD Param = 1);
  15 Node № 15; X_Level = 1; Mark = 1; (ADD Param = 1);
  16 Node № 16; X_Level = 1; Mark = 1; (ADD Param = 1);
  17 Node № 17; X_Level = 1; Mark = 1; (ADD Param = 1);
  18 Node № 18; X_Level = 1; Mark = 1; (ADD Param = 1);
  19 Node № 19; X_Level = 1; Mark = 1; (ADD Param = 1);
  20 Node № 20; X_Level = 1; Mark = 1; (ADD Param = 1);
  21 Node № 21; X_Level = 1; Mark = 1; (ADD Param = 1);
  
```

NewBase

Предложенные для исследования структуры графов-деревьев ИЗОМОРФНЫ

Это было доказано, когда в качестве НОВОГО корня для изоморфной реконфигурации структуры 2_Тее (режим "Ветка за веткой") была взята ее вершина, занимающая Y-уровень №3.

OK

Рис. 1

Все необходимые для решения настоящей задачи операции, равно как и реализующая этот метод, — часть их базовой (допускающей возможность пополнения) совокупности, интегрированной в структуре созданного комплекса

программных средств. Указанная совокупность обеспечивает возможность построения «с нуля» произвольной структуры графа-дерева, осуществления ее всевозможных преобразований, получения ряда полезных в практическом отношении характеристик, постоянную оперативную визуализацию на экране монитора всех происходящих событий и их результатов, а также минимальное участие в этом процессе пользователя. Конкретно использованные для решения нашей задачи операции выделены крестиком ('+') и более темным цветом соответствующих кнопок.

Для визуализации структур деревьев, используется часто встречающийся на практике метод. При этом условными порядковыми номерами вершин (Node №*) являются значения Y -уровней, которые занимают вершины на вертикальной (направление сверху вниз) Y -оси плоскости экрана монитора. Выбор такой формы графического представления деревьев объясняется следующими причинами. Во-первых, лаконичностью и простотой изображения, существенно облегчающей понимание всех деталей и особенностей взаимосвязи вершин. Во-вторых, возможностью постоянного присутствия на экране совокупности связанных с вершинами и используемых в приложениях параметров, в-третьих, удобством реализации scroll-режима для просмотра деревьев с большим числом вершин. В-четвертых, совпадением количества вершин графа с общим числом визуально отображаемых Y -уровней. В-пятых, идентичностью номеров соответствующих Y -уровней с порядковыми номерами вершин (Node №*) в вертикальной плоскости экрана, а, значит, и с совокупностями связанных с ними параметров (Mark = *, Degree = *, Add.Param.- и др.). Нумерация вершин и Y -уровней начинается с «0» в связи с тем, что:

1) корень дерева является вершиной со «специфической» ролью и свойствами, поэтому его целесообразно каким-то образом выделить среди остальных вершин, если степень корня равна единице, в ряде случаев его удобно трактовать как лист;

2) в большинстве языков программирования нумерация элементов создаваемых программными средствами в памяти ПК и динамически распределяемых массивов данных начинается с цифры «0».

Содержащиеся в строках параметров значения X -уровней ($X_Level = *$), стоящие за условными порядковыми номерами (Node №*), означают место (позицию) размещения вершин на общей горизонтальной X -оси (направление слева направо) при их виртуальном и строго вертикальном перемещении к верхней кромке плоскости экрана монитора.

В излагаемом методе совокупность связанных с каждой вершиной данных, в том числе включающих используемые в различного рода приложениях «вспомогательные» параметры, представляется структурированной символьной строкой постоянной длины r . Временно в данной версии комплекса $r = 32$. Длина r , в принципе, может изменяться, ориентируясь на специфику исследуемых совокупностей деревьев и решаемые задачи. При этом безызыбочная совокупность данных всего N -вершинного дерева хранится в простейшего вида бинарном файле FileName.tre объема Nr , что, при необходимости, позволяет максимально оперативно восстановить любой из видов его формализованного представления и получить все данные, необходимые как для реализации какого-либо из ныне существующих методов анализа изоморфности, так и при решении различного рода задач прикладного характера. На прилагаемых рисунках в качестве примера приведены только некоторые из параметров, так как их общее число может быть произвольным.

Настоящая версия программных средств обеспечивает работу с деревьями, количество вершин N которых не превышает значения 65536. Никакие ограничения на взаимосвязь между максимально допустимыми значениями X и Y уровней структуры дерева не накладываются, а единственное, требующее выполнения условие состоит в том, что общее число вершин N исследуемых деревьев не должно превышать указанного выше значения. Поскольку созданный комплекс программных средств предназначен для решения и иных задач, то данное условие учитывает и такой, своего рода «предельный» случай, когда N -вершинное дерево имеет единственный лист и все входящие в его состав вершины являются элементами единственного существующего на дереве пути. При этом есть весьма веские предпосылки говорить о том, что максимально допустимое количество X -уровней анализируемых деревьев можно ограничить значением 256.

Следует отметить, что заложенные в существующей версии программных средств возможности, будучи воплощенными в ее последующей модификации, позволят сделать реальной работу с деревьями, количество вершин N которых сможет достигать значений $N^* = 65536 * 256^k$ ($k = 1, 2, 3, 4, \dots$).

Возьмем в качестве контрольного G^1 любое $N1$ -вершинное ($2 < N1$) дерево и перенумеруем его вершины $\{g_n^1\}_{n=0}^{N1-1}$ в произвольном порядке натуральными числами n ($0 \leq n \leq N1 - 1$). Назовем эти числа «личными» номерами вершин g_n^1 , степени которых обозначим d_n^1 . На основе G^1 построим множество \bar{G}^1 всех изоморфных ему графов и разместим «личные» номера вершин внутри изображающих их прямоугольников. Разбивая \bar{G}^1 на $N1$ классов $\left\{G_n^1\right\}_{n=0}^{N1-1} \left(\bar{G}^1 = \bigcup_{n=0}^{N1-1} G_n^1\right)$, где корнем всех входящих в G_n^1 элементов является одна и та же вершина $g_n^1 \in G^1$, будем говорить, что G_n^1 порожден g_n^1 с «личным» номером n . Говоря о $\{G_n^1\}_n$, важно помнить, что для ряда значений i и j ($i \neq j, 0 \leq i, j \leq N1 - 1$) условие $G_i^1 \cap G_j^1 = \emptyset$ может не выполняться. Более того, возможны ситуации, когда $G_i^1 \equiv G_j^1$. Так как деревья G^0 и G^1 выбираются произвольно, каждое из них может обладать указанной особенностью.

«Личные» номера, автоматически присваиваемые вершинам в момент появления в структуре дерева и указываемые в полях обозначающих их прямоугольников, носят сугубо формальный характер. При этом изначально сформированный порядок следования этих номеров сверху вниз может нарушаться, а часть из них может «исчезать» в результате проведения над деревом различного рода преобразований структуры. Отмеченный факт иллюстрирует рис. 1.

«Личные» номера не несут никакой информационно-смысловой, полезной для решения поставленной задачи нагрузки, и приводятся на рисунках исключительно для удобства пользователя в качестве визуальных идентификаторов вершин. Навсегда «привязанные» к конкретным вершинам, они остаются неизменными после различного рода модификаций структуры графов, не учитываются ни при реализации предложенного метода решения сформулированной задачи, не используются ни в качестве параметров строк, формализующих конструируемые пути на эталонном и тестируемом графах, ни в последующем сравнении пар путей

при вынесении вердикта об изоморфности структур графов G^0 и G^1 . Для их изменения предназначена специально разработанная и включенная в состав комплекса операция, которая активизируется нажатием кнопки с надписью «10. Перенумеровать вершины». Воспользоваться кнопкой можно в любой момент, и результатом такого действия станет присвоение вершинам строго упорядоченных (последовательно, сверху-вниз по Y-оси экрана монитора) новых «личных» номеров, представляющих собой натуральные числа $0, 1, 2, \dots, R - 1$, которые при новом общем количестве вершин R заменят ранее присутствовавшие на рисунке значения.

В основу разработанного метода генерации изоморфных структур положено одно из известных определений дерева как графа, на котором существует единственный путь, соединяющий любые пары вершин и приведенные ниже свойства.

Обозначим L^1 общее количество листьев контрольного графа G^1 , а L_n^1 ($1 \leq n \leq N1 - 1$) — число листьев у деревьев класса G_n^1 . Объединим в отдельные группы размеров D_q^1 вершины G^1 с идентичным (разным для разных групп) значением степени q . ($1 \leq q$). Тогда:

- 1) изоморфное преобразование G^1 в любой элемент из $\left(\overline{G}^1 = \bigcup_{n=0}^{N1-1} G_n^1 \right)$

оставляет неизменным и число инцидентных каждой вершине ребер, и размеры $\{D_q^1\}_q$ всех групп вершин с конкретными значениями степени q ;

- 2) количество висячих вершин L_n^1 у всех представителей класса G_n^1 является величиной постоянной;

- 3) для любых значений i и j ($0 \leq i, j \leq N1 - 1, i \neq j$) и классов G_i^1 и G_j^1 величина $\delta_{ij} = |L_i^1 - L_j^1|$ удовлетворяет условию $0 \leq \delta_{ij} \leq 1$, причем $\delta_{ij} \equiv 1$ только в том случае, когда в одном из этих классов роль корня исполняет лист графа G^1 ;

- 4) при каждом фиксированном i все представители класса G_i^1 имеют одно и то же максимальное количество X-уровней.

Эти принципиально важные моменты стали основой при создании и программной реализации четырех различных вариантов изоморфной реконфигурации структуры деревьев, которые должны осуществляться при одной и той же вершине, исполняющей роль нового корня. На рис. 1 указанные варианты реконфигурации объединены общим названием «Изоморфная реконфигурация структуры графа при замене корневой (*ROOT*) вершины».

Выбор и последующая реализация требуемого варианта реконфигурации осуществляются нажатием соответствующей кнопки, входящей в группу под общим заголовком «Реконфигурировать исходную структуру графа (для 'НОВОЙ' ROOT вершины)». Для выполнения указанной операции пользователю необходимо лишь промаркировать с помощью левой кнопки манипулятора вершину, на которую возлагается роль нового корня, выбрать соответствующий вариант стратегии формирования структуры нового дерева и активизировать кнопку «Показать результат выполнения операции на дереве». Операция применима к любой вершине дерева, кроме исполняющей роль корня, но ее применение к указанному объекту становится возможным, если в результате предшествующих действий его «структурный статус» был изменен.

Количество последовательно осуществляемых над исходной структурой N -вершинного дерева операций изоморфной реконфигурации ничем не ограничено, но ее многократное выполнение может привести к появлению структур, «топологически идентичных» полученным ранее.

Сказанное выше позволяет сделать вывод, что при решении вопроса изоморфности эталонному дереву G^0 из совокупности предлагаемых на рассмотрение структур или структур, получаемых из контрольного G^1 в результате его изоморфной реконфигурации, должны отбираться только те, которые обладают следующей совокупностью свойств, имеют:

- 1) одинаковое с G^0 количество вершин;
- 2) как минимум, одну вершину, степень которой совпадает со степенью корневой вершины G^0 ;
- 3) такое же, как у G^0 , количество групп вершин с аналогичными для конкретных значений степени q размерами D_q^0 ;
- 4) идентичное с G^0 число вершин, степень которых равна единице. Это условие позволяет учесть случаи, когда корнем G^0 или G^1 является вершина, которая сохранила такую степень с момента создания дерева или, будучи листом, стала корнем в результате ряда операций;
- 5) значение максимальной длины пути {лист g_n^1 } $\leftrightarrow \dots \leftrightarrow g_r^1 \leftrightarrow \dots \leftrightarrow ROOT$, совпадающее с аналогичной характеристикой G^0 .

При большом количестве вершин предварительный отбор удовлетворяющих этим условиям структур позволяет существенно сократить число подлежащих анализу объектов и, как следствие, общее время получения результата решения задачи. Так в приведенном примере данные дерева G^0 брали из файла 1_Tree.tre, а G^1 — из файла 2_Tree.tre. При этом, если не учитывать свойство 5, общее число структур, подлежащих тестированию на изоморфность G^0 , не превысит значения 4.

Структуры обоих деревьев содержали 21 вершину, и их объемы равнялись 672 (21 x 32) байта. Применялся первый вариант конструирования четырех представителей структур, изоморфных контрольному графу-дереву 2_Tree.tre, которое, в свою очередь, является результатом различного рода преобразований. Об этом можно легко судить по порядку следования «личных» номеров вершин, изначально упорядоченных сверху вниз.

Вершины дерева 2_Tree.tre, исполняющие роль корней его четырех изоморфных аналогов, имеют единое для всех значение степени (4) и «личные» номера 11, 9, 0, 17. Единый для всех четырех случаев и примененный к 2_Tree.tre способ изоморфной реконфигурации, (см. рис. 1) располагается под кнопкой «7. Реконфигурировать исходную структуру графа (для «НОВОЙ» ROOT вершины)» и носит название «Ветка за веткой». Цель такого решения — внесение дополнительного элемента общности и некоторой универсальности, обеспечение лучшего понимания сути излагаемого метода и логичности производимых действий, способствовало дополнительному упрощению решения сформулированной задачи и снижению общих затрат ресурсов. Четыре изоморфных представителя, полученные в результате операций изоморфной реконфигурации структуры графа-дерева 2_Tree.tre после конкретизации их новых корней приведены на рис. 2.

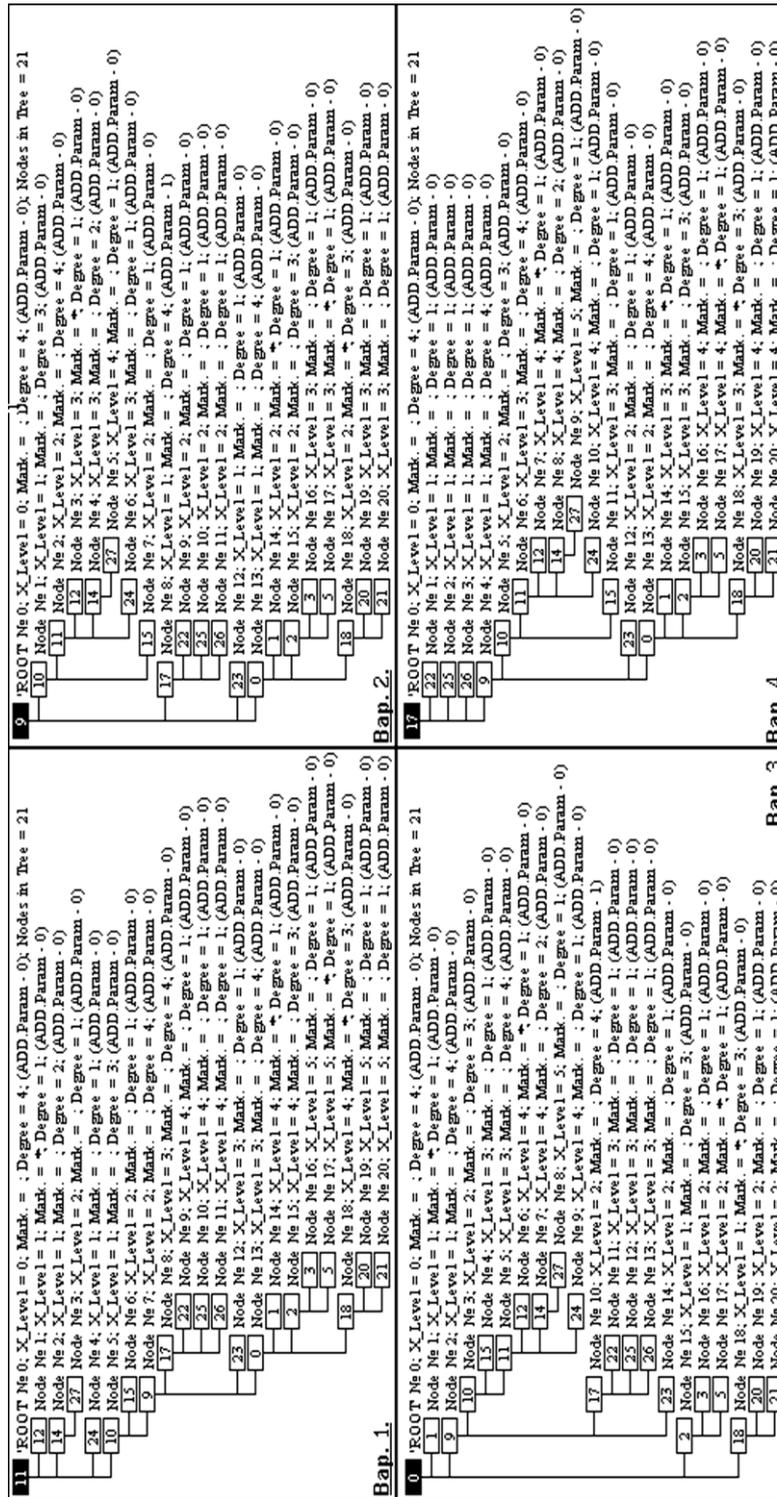


Рис. 2

Как следует из определения графа-дерева, если контрольное и эталонное деревья удовлетворяют условиям 1–5, их корни имеют одинаковую степень m и при этом каждый путь контрольного дерева, который начинается листом g_n^1 (или корнем $ROOT$) и заканчивается корневой $ROOT$ -вершиной (или листом g_n^1),

имеет единственный аналог на эталонном дереве (и наоборот), то структуры указанных графов-деревьев изоморфны. Справедливость этого вытекает из возможности преобразования излагаемого в настоящей статье метода в известный метод сравнения, более ресурсозатратный как по реализации, так и по использованию. Вполне очевидно, что сформулированная закономерность справедлива, если для установления факта изоморфности эталонное и контрольное деревья поменять ролями.

Поэтому одной из целей, преследуемой при практической реализации предлагаемого метода решения задачи, также является разработка средств максимально эффективной селекции единственных представителей целенаправленно отбираемых классов $\{G_i^1\}_i$, структура которых изоморфна контрольному дереву G^1 , а также максимально быстрого построения и последующего сравнения существующих на них путей.

В качестве основного метода контроля идентичности пар путей, принадлежащих различным, тестируемым на изоморфность деревьям, выбран используемый в языках программирования метод лексикографического сравнения символьных строк, которые в данном случае использовались как средства формализованного представления самих путей.

Вспоминая, что нынешняя версия программных средств рассчитана на работу с деревьями, общее количество вершин которых не превышает значения 65536, смысл использования подобного представления заключается в следующем. Легко видеть, что формальное назначение вершинам G^0 и G^1 произвольных «личных» номеров делает невозможной разработку метода, который, для облегчения решения задачи, позволил бы идентифицировать пары принадлежащих различным деревьям вершин. В то же время выяснилось, что для информативно-полноценного описания пути $\{\text{лист } g_n^1\} \leftrightarrow \dots \leftrightarrow g_r^1 \leftrightarrow \dots \leftrightarrow \text{ROOT}$, который начинается с листа g_n^1 (или ROOT -корня), имеет длину W_n^1 и порядок следования вершин в котором заведомо определяется алгоритмом его построения, достаточно иметь формализующую путь символьную строку, состоящую из $3W_n^1$ компонентов char -типа. Или же для характеристики каждой, входящей в путь вершины, требуются всего три соответствующим образом кодируемые char -символа.

Поэтому согласно свойству 5 при наличии эталонного (G^0) и контрольного (G^1) деревьев для учета различных, входящих в их структуры листьев $\{g_q^*\}_q$ ($*$ = 0 или $*$ = 1) и, значит, наличия на каждом из них «своих» путей различной длины $\{W_q^*\}_q$, связанные с каждым из деревьев строки должны иметь единую основу. В данном случае для каждого из графов, роль такой основы играет своя «универсальная» постоянная длина $L^* = \max_q(W_q^*)$, рассчитанная на описание

пути максимально допустимой длины, которая, как легко видеть, должна совпадать с утроенным максимальным значением количества X -уровней соответствующего графа. Более того, для возможности сравнения таких строк значение L^* должно быть одним и тем же, как для G^0 , так и для G^1 . Отметим также, что при различной длине принадлежащих одному и тому же дереву путей ряд параметров «хвостовой» части формализующих их строк оказываются незадействованными, тем самым частично снижая эффективность использования оперативной памяти ПК.

Работа программы организована таким образом, что в процессе решения задачи все структуры изоморфных аналогов графа G^1 , отобранные согласно указанным ранее правилам, из общей совокупности получаемых в результате использования операции изоморфной реконфигурации, последовательно выводятся на экран в главном информационном окне комплекса с заголовком «Окно визуализации структуры рабочего варианта графа». Вывод структур осуществляется в порядке их генерации программой. Имеется два режима работы.

1. После вывода структуры и анализа аналога G^1 на изоморфность G^0 программа, выдав соответствующее сообщение, останавливается, ожидая реакции на него пользователя и только потом переходит к анализу очередного аналога.

2. Программа сама, последовательно, без прерываний своей работы, перебирает все отобранные структуры. В большинстве случаев это происходит достаточно быстро и, как правило, не предоставляет пользователю возможности детального знакомства с их внешним видом и присущими им топологическими особенностями.

Программа завершает работу либо после получения положительного ответа на вопрос изоморфности некоторого изоморфного аналога G^1 и G^0 , либо когда придет к негативному ответу после анализа всех выбранных структур. В любом случае финальный результат тестирования графов на изоморфность формулируется без участия пользователя самой программой в виде соответствующего вида сообщения, образец которого при положительном ответе приведен на рис. 1.

В завершение отметим ряд важных в практическом отношении моментов.

1. В общем случае угадать порядок выбора вершин контрольного дерева из общей совокупности удовлетворивших ранее сформулированным условиям и уже отобранных в качестве новых корней для генерации подлежащих тестированию структур, для максимально быстрого получения ответа на интересующий нас вопрос практически невозможно. Преследуя ту же цель, не менее проблематично принудительно уменьшить число членов этой совокупности, если общее количество вершин, пригодных для роли корней в генерируемых структурах, достаточно велико. Это вынуждает работать отдельно с каждым из тестируемых деревьев и в том порядке, который установлен в самой процедуре генерации изоморфных структур. Тем не менее настоящий метод уже обладает свойством, позволяющим, при возможности, все-таки дополнительно сократить число подлежащих тестированию структур, которые, как легко видеть, принадлежат различным G_n^1 . Суть такого сокращения заключается в следующем.

При лексикографическом сравнении строк, формализующих пары путей сформированных на эталонном и тестируемых деревьях, принципиальное значение имеет не только их символическое наполнение, но также реальная и максимальная длины самих строк. Как указано ранее, последняя совпадает с максимальным числом X -уровней графа, поэтому определяется каждый раз процедурой, изоморфной реконфигурации в процессе генерации для новой коревой вершины g_n^1 дерева, предназначенного тестированию.

Поэтому заблаговременно определенные программой и различные по величине значения максимальных длин строк эталонного и тестируемого графов, «автоматически» становятся сигналом о нецелесообразности анализа изоморфности экземпляра дерева G_n^1 , изоморфного G^1 и G^0 , и программа, отказавшись от этого, перейдет к генерации и анализу очередного тестируемого варианта графа, сокращая общее время своей работы. Именно этот факт объясняет необходимость введения свойства отбора 5.

Использование свойства 5 исключит из приведенной на рис. 2 совокупности и вариант 2, нарушающий условие идентичности значений максимальной длины символьных строк.

К сказанному целесообразно добавить, что при реализации процедуры изоморфной реконфигурации решить вопрос порядка выбора вершин-корней из уже выявленной их совокупности можно несколькими способами, а именно: согласно их непосредственному расположению на Y -уровнях контрольного дерева (снизу вверх, сверху вниз), с помощью датчика случайных чисел или путем соответствующих программных установок. Порядок использования в примере отобранных в G^1 вершин-корней осуществлялся по принципу снизу вверх согласно их расположению на Y -уровнях контрольного дерева `2_Tree.tre`.

2. Поскольку предлагаемые к проверке на изоморфность структуры «равноценны» в смысле назначения какой-либо из них на роль эталонной, поступить можно следующим образом. Если эти структуры удовлетворили приведенным ранее условиям и им вынесен вердикт о целесообразности проверки, то предварительно средствами комплекса, в каждой из них надо найти минимальное количество вершин (Z_1 в условно первой структуре и Z_2 в условно второй) с идентичным, общим для обоих деревьев значением степени d . Определив $Z = \min\{Z_1, Z_2\}$, назначить дерево с таким количеством вершин в качестве контрольного, а второе, автоматически обретающее статус эталонного, разово модифицировать процедурой изоморфной реконфигурации, выбрав в качестве его нового корня любую вершину степени d . Затем, определив модифицированное дерево как эталонное G^0 , реализовать процедуру анализа изоморфности с помощью разработанного метода.

3. Существующая в настоящее время версия программных средств обеспечивает работу с деревьями, максимальное количество вершин которых не может превышать значение $N = 65536$. Указанное значение носит временный характер, поскольку в созданных программных средствах заложена возможность, позволяющая, при необходимости и соответствующей доработке, доводить общее количество вершин исследуемых деревьев до значений $K1 = 65536 * 256^k$ ($k = 1, 2, 3, \dots$), а число X -уровней — до $X1 = 256 * 256^q$ ($q = 1, 2, 3, \dots$). При этом вполне разумным выглядит требование к выполнению условия $X1 < K1$.

4. При незначительной доработке, помимо ранее упомянутых двух вариантов матрицы смежности, программа сможет формировать и записывать в файл и матрицу инцидентности, и все другие виды формализованного представления структуры дерева, которые, в случае необходимости, будут использованы для решения различного рода прикладных задач.

5. Главной особенностью изложенного в настоящей статье метода является то, что для описания, построения, визуализации, различного рода модификаций структуры любого N -вершинного ($2 < N$) дерева и решения сформулированной задачи разработанным методом не требуется иметь дело ни с одним из созданных видов формализованного представления деревьев. Вместо этого, при необходимости указания значений «личных» номеров вершин, для каждого дерева, с помощью средств комплекса достаточно создать исходный массив данных, содержащий только $4N$ значений `char`-типа параметров, что во много раз меньше их количества, обычно хранимого в памяти ПК при использовании различного рода СФПД. Если такой необходимости нет, то всего $2N$ значений. Такой массив, как и сейчас, будет храниться и в памяти ПК и в простейшего вида бинарном файле вида `1_Tree.tre` и `2_Tree.tre`, но не будет содержать совокупности дополнительных, нужных лишь для различного рода приложений параметров, часть которых на рис. 1 и рис. 2 приведена в строках, находящихся непосредственно за изображающими вершины прямоугольниками.

Заключення

Разработаны метод и реализующие его программные средства оценки изоморфности пары неориентированных деревьев с произвольной структурой и нумерованными вершинами, общее число которых не превышает значения 65536. Основой разработки является созданный метод изоморфной реконфигурации, который для каждой выбираемой в качестве нового корня вершины позволяет получить четыре различных варианта структур, изоморфных исходной структуре дерева.

При практической реализации метод не опирается ни на одно из известных средств формализованного описания (представления) структуры деревьев и не требует их использования при выполнении над последней какого-либо рода операций преобразования. Он также исключает необходимость освоения и последующего использования в работе стандартизированных языков описания графов, специализированных программ построения их структуры, визуализации и т.д.

Все данные N -вершинных деревьев, без учета дополнительных, допускающих пополнение параметров, используемых в качестве справочной информации и для решения различного рода прикладных задач, хранятся в простейшего вида бинарном файле общего объема $4N$ char-типа символов.

При необходимости, два различных варианта матрицы смежности могут быть сформированы программой в конце работы и записаны в файлы с расширениями *.sam (классический вариант) и *.tam (усеченный вариант, содержащий только наддиагональные элементы матрицы).

Метод способствует многократному уменьшению общих затрат ресурсов и времени решения задачи, а соответствующая доработка его нынешней версии сделает возможной решение задачи изоморфизма деревьев с практически неограниченным числом вершин и произвольной ориентацией ребер.

О.І. Іванешкін

ВИРІШЕННЯ ПИТАННЯ ІЗОМОРФІЗМУ НЕОРІЄНТОВАНИХ ДЕРЕВ МЕТОДОМ ГЕНЕРАЦІЇ ІЗОМОРФНИХ СТРУКТУР

Розроблено метод і реалізуючі його програмні засоби оцінки ізоморфності пари неорієнтованих дерев з довільною структурою і нумерованими вершинами, загальне число яких не перевищує значення 65536. Зв'язок користувача із засобами здійснюється на візуально-ментальному (аналогічно грі «LEGO») рівні спілкування, що мінімізує його участь в процесі виконання завдання. Основою розробки є створений метод ізоморфної реконфігурації, який для кожної обраної в якості нового корня вершини дозволяє отримати чотири різні варіанти структур, ізоморфних початковій структурі дерева. Для практичного застосування засоби не вимагають використання матриць суміжності або будь-яких інших засобів формалізованого опису (подання) структури дерев, а також їх подальшого застосування при виконанні над останньою різного роду операцій перетворення. Вони також виключають необхідність освоєння і використання в роботі стандартизованих мов опису графів, спеціалізованих програм побудови їх структури, візуалізації тощо. За бажанням користувача, два варіанти матриці суміжності можна сформувати програмою в кінці роботи і записати в файли з розширеннями * .sam, умовно, класичний (classic) варіант і *.tam, умовно, усічений (truncated) варіант, що включає елементи лише верхньої частини матриці. Всі дані N -вершинного дерева (без урахування додаткових параметрів), що потрібні для вирішення завдання, зберігаються в файлі бінарного типу, за-

гального обсягу $4N$ символів char-типу. Метод сприяє зменшенню загальних витрат ресурсів і часу виконання завдання і відповідне доопрацювання нині існуючої версії засобів зробить можливою роботу зі структурами дерев, що мають практично необмежену кількість вершин і довільну орієнтацію ребер.

Ключові слова: граф, дерево, неорієнтоване дерево, ізоморфізм.

A.I. Ivaneshkin

THE SOLUTION OF THE QUESTION OF THE ISOMORPHISM OF NON-ORIENTED TREES BY THE METHOD OF GENERATING ISOMORPHIC STRUCTURES

The method and its software implementation tools for assessing the isomorphism of a pair of non-oriented trees with an arbitrary structure and unnumbered nodes, the total number of which does not exceed the value of 65536, are developed. The user communicates with the tools at the visual-mental (similar to the «LEGO» game) communication level, minimizing his participation in the process of solving the problem. The basis of development is the created method of isomorphic reconfiguration, which, for each node selected as a new root, allows to obtain four different variants of structures isomorphic to the original tree structure. For practical use, the tools do not require the use of adjacency matrices or any other means of a formalized description (representation) of the structure of trees and their subsequent application when performing its various transformations. They also eliminate the need for mastering and using standardized graph description languages, specialized programs for constructing their structure, visualization, etc. If the user wishes, two variants of the adjacency matrix can be generated by the program at the end of the work and written to files with the extensions *.cam, conditionally, the classic option and *.tam, conditionally, the truncated option containing elements only of the upper part of the matrix. All the data of the N -node tree required for solving the problem (without taking into account additional parameters) is stored in a binary file type, with a total volume of $4N$ char-type characters. The method helps to reduce the total cost of resources and time to solve the problem and the corresponding refinement of the current version of the tools will make it possible to work with tree structures that have an almost unlimited number of nodes and arbitrary orientation of the edges.

Keywords: graph, tree, undirected tree, isomorphism.

1. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. М.: Наука, 1990. 384 с.
2. Евстигнеев В.А., Касьянов В.Н. Теория графов: алгоритмы обработки деревьев. Новосибирск: Наука, 1994. 360 с.
3. Просолупов Е.В. Курс лекций по дискретной математике. Часть 3. Теория алгоритмов и теория графов. СПб.: Изд-во С.-Петербур. ун-та, 2014. 84 с.
4. Bollobas B. Modern graph theory. Springer, 2013. 394 p.
5. Gross J.L., Yellen J., Zhang P. Handbook of graph theory. CRC Press, 2014. 1632 p.
6. McKay B.D. Practical graph isomorphism. *J. Symbolic Computation*. 2014, **60**, P. 94–112.

Получено 27.01.2019