Павлов Дмитрий Алексеевич

КАНОНИЧЕСКИЙ КОД ДЕРЕВА

Аннотация

Канонический код – это способ записи графа, инвариантный относительно изоморфизма. В статье вкратце объясняется значимость канонических кодов и показывается связь задачи вычисления канонического кода с другими задачами, известными из курса теории алгоритмов. Приводится алгоритм вычисления канонического кода для дерева с помеченными вершинами и рёбрами, который работает за линейное время от размера дерева. Алгоритм основан на известном алгоритме проверки изоморфизма двух деревьев.

Ключевые слова: канонический код, каноническая нумерация, изоморфизм графов, изоморфизм деревьев.

ВВЕДЕНИЕ

Определим сначала, что является каноническим кодом произвольного графа (graph canonical code). Таким кодом называется уникальная строка, которая не зависит от порядка нумерации вершин. У изоморфных графов одинаковые канонические коды, у неизоморфных – разные.

Вершины и рёбра графа могут быть снабжены дополнительными метками (цветами). В этом случае канонический код включает в себя цветовые идентификаторы.

Имея в руках канонические коды графов, можно определять их изоморфизм простым сравнением строк, не делая дорогостоящей процедуры проверки изоморфизма (graph isomorphism, GI).

Например, как найти дубликаты в множестве N графов? Вычислить N канонических кодов, отсортировать их и последовательно сравнить. Это гораздо выгоднее, чем делать $N\cdot(N-1)/2$ проверок изоморфизма. Можно сказать, что канонический код – это хеш-код без промахов.

Канонический код может выглядеть поразному: как матрица смежности, развёрнутая в строку, или как маршрут обхода графа в ширину или в глубину. Какое представление выбрать, не столь важно. Задача в любом случае сводится к вычислению канонической нумерации вершин (canonical numbering, canonical labeling). Каноническая нумерация – это нумерация (перестановка) вершин графа, гарантирующая, что изоморфные графы будут пронумерованы одинаково. Получив нужную перестановку, уже можно без труда построить канонический код, например, по обходу графа в глубину: начать обход с вершины под номером 1 и при ветвлении выбирать ту вершину, «канонический номер» которой меньше.

Наличие полиномиального алгоритма для задачи канонической нумерации означало бы, что задача GI определения изоморфизма графов полиномиально разрешима. На данный момент для задачи GI полиномиального алгоритма не найдено, но и не доказана NP-полнота (в отличие от известной задачи изоморфизма подграфу, subgraph isomorphism). По некоторым признакам можно предполагать, что GI не является NP-полной [1].

В ряде частных случаев [2, 3, 4] существуют полиномиальные алгоритмы проверки изоморфизма и канонической нумерации. Мы рассмотрим простейший случай, когда граф является деревом.

В книге Ахо, Хопкрофта и Ульмана [5] приводится алгоритм для определения изоморфизма двух деревьев (tree isomorphism), работающий за линейное время от количе-

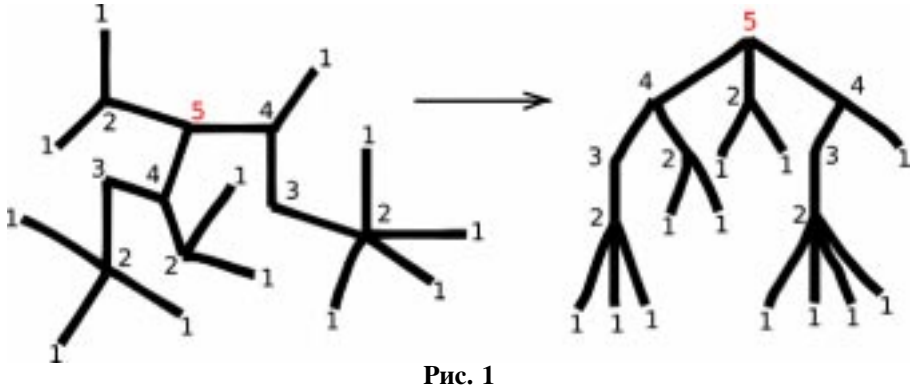

Рис. 1

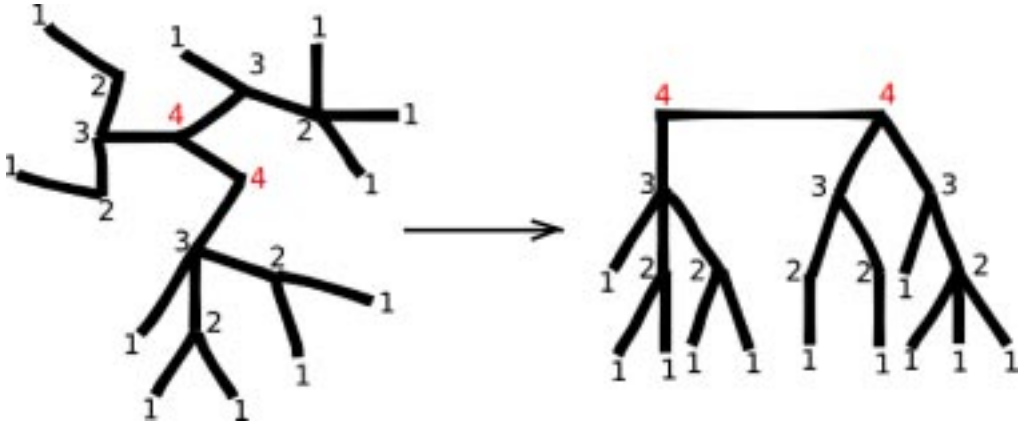

Рис. 2

ства вершин. На его основе нетрудно составить алгоритм канонической нумерации вершин дерева, работающий за то же линейное время. Алгоритм состоит из трёх этапов, изложенных в трёх следующих параграфах.

ЭТАП 1. ВЫДЕЛЕНИЕ КОРНЯ

Отрубим от дерева все листья. Получится дерево меньшего размера. Будем повторять процедуру до тех пор, пока не останется одна вершина (center) или две вершины, соединённых ребром (bicenter).

Если осталась одна вершина, мы делаем её корнем и располагаем остальные вершины по уровням, считая от корня (рис. 1).

Если осталось ребро, мы располагаем остальные вершины по уровням, считая от этого ребра, обе вершины которого помещаются на первый уровень (рис. 2).

Мы разделили N вершин на K уровней, так что у каждой вершины на уровне $k > 1$ есть «родитель» из уровня $k - 1$. Вспомним ещё, что вершины и рёбра снабжены цветами. В нашем примере они будут чёрными и серыми (рис. 3).

ЭТАП 2. ПОУРОВНЕВАЯ СОРТИРОВКА

Для алгоритма нужно задать условный порядок на цветах. Пусть серый цвет будет старше чёрного. Введём обобщённый код вершины, состоящий из цветового кода ребра, которое соединяет её с родителем, и из цветового кода самой вершины.

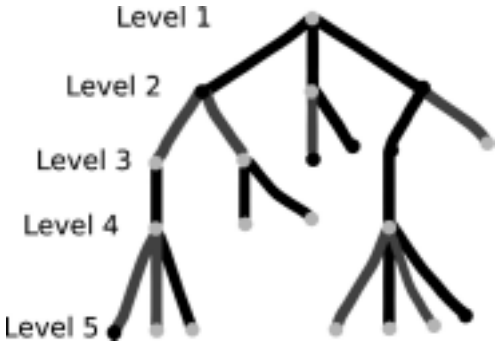

Рис. 3

Сгруппируем вершины нижнего уровня, имеющие одинаковый обобщённый код, и присвоим каждой группе ранг, соответствующий старшинству цветов. Цвет ребра имеет больший приоритет, чем цвет вершины. Ранг 0 считается самым старшим (рис. 4).

Рис. 4

Составим обобщённые коды родителей этих вершин, которые расположены на предыдущем уровне. Кроме цветовых кодов ребра к родителю и самой вершины, включим в обобщённый код отсортированный список рангов её детей. Ранги детей имеют меньший приоритет, чем цветовые коды вершины и ребра (рис. 5).

Рис. 5

Повторяем описанную процедуру (рис. 6) и ещё раз (рис. 7), пока не дойдём до уровня 1. В нашем случае на первом уровне одна вершина, поэтому её ранг нулевой, в случае двух вершин на первом уровне нужно сравнить их обобщённые коды и присвоить им ранги 0 и 1 (или 0 и 0).

Рис. 6

Рис. 7

ЭТАП 3. НУМЕРАЦИЯ

Начиная с уровня 1, присваиваем вершинам возрастающие номера в порядке возрастания их рангов. В случае совпадения рангов на уровне порядок присваивания номеров может быть любым, так как вершины с одинаковым рангом в некотором смысле идентичны. Выражаясь математически, эти вершины лежат на одной орбите [6], то есть существует автоморфизм графа, переводящий одну вершину в другую (рис. 8).

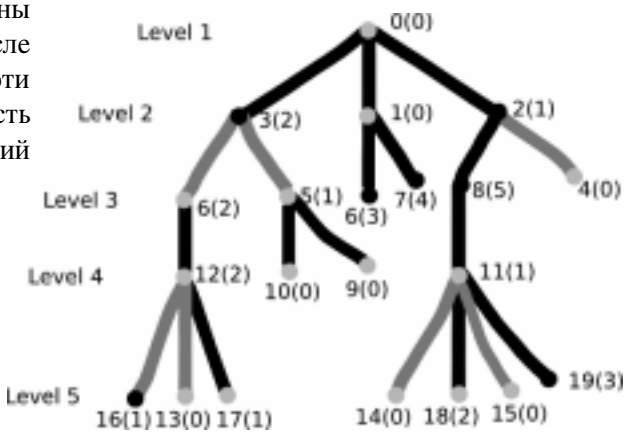

Рис. 8

ПОСТРОЕНИЕ КАНОНИЧЕСКОГО КОДА

Как было сказано, строить канонический код можно многими способами. Приведём пример канонического кода, построенного при обходе дерева в глубину (рис. 9).

Рис. 9

Для построения такого кода не нужен даже третий этап предыдущего алгоритма (нумерация). Мы начинаем обход с корня и для каждой посещённой вершины обходим её детей в порядке старшинства. Для определения старшинства детей хватит «внутриуровневых» кодов, полученных на этапе 2.

АНАЛИЗ СЛОЖНОСТИ

Этап 1, очевидно, требует $O(N)$ операций. Нетрудно также понять, что на этапе 2 мы, двигаясь от последнего уровня к первому, проходим все вершины по одному разу. Затруднения может вызвать только процесс «ранжирования» вершин на уровне, но при использовании поразрядной сортировки [7] он занимает время, пропорциональное количеству вершин, и суммарно по уровням получается тоже $O(N)$ операций. Этап 3 и построение самого кода также занимают линейное время.

КАНОНИЧЕСКАЯ НУМЕРАЦИЯ В ОБЩЕМ СЛУЧАЕ

Австралийский профессор Brendan McKay написал библиотеку на C под названием Nauty [8]. Она предназначена для поиска групп автоморфизмов произвольных графов; одним из её «побочных» результатов является каноническая нумерация вершин.

Nauty работает очень быстро. Она не предоставляет возможности задать цвета рёбер, но при необходимости всегда можно добавить в граф псевдо-вершины таким образом, что каноническая нумерация получившегося графа без учёта цветов рёбер будет являться канонической для исходного графа.

Литература

1. *Johannes Kobler, Uwe Schoning, Jacobo Toran.* The Graph Isomorphism Problem: Its Structural Complexity. Birkhauser, 1993.

2. *J. Hopcroft, J. Wong. A Linear.* Time Algorithm for Isomorphism of Planar Graphs, Proceedings of the Sixth Annual ACM Symposium on Theory of Computing, 1974, pp. 310–324.

3. *E.M. Luks.* Isomorphism of Graphs of Bounded Valence Can Be Tested in Polynomial Time, Proc. 21st IEEE FOCS Symp., 1980, pp. 42–49.

4. *L. Babai, D.Y. Grigoryev, D.M. Mount.* Isomorphism of Graphs With Bounded Eigenvalue Multiplicity, Proceedings 14th ACM Symposium on Theory of Computing, 1982, pp. 310–324.

5. *А. Ахо, Дж. Хопкрофт, Дж. Ульман.* Построение и анализ вычислительных алгоритмов. М.: «Мир», 1979.

6. http://mathworld.wolfram.com/GroupOrbit.html

7. *Дональд Кнут.* Искусство программирования, том 3. Сортировка и поиск = The Art of Computer Programming, vol.3. Sorting and Searching. 2-е изд. М.: «Вильямс», 2007.

8. http://cs.anu.edu.au/~bdm/nauty/

Abstract

Canonical code is the notation for the graph structure, which is invariant to isomorphism. The article briefly illustrates the significance of canonical codes in general, and reveals some connection of the canonical code building to other problems common to computer science. A linear-time algorithm for calculating the canonical code of a colored tree is presented. The algorithm is based on widely known procedure of tree isomorphism detection.

Павлов Дмитрий Алексеевич,
аспирант кафедры прикладной математики ФМФ СПбГПУ,
dmitry.pavlov@gmail.com