АЛГОРИТМЫ АНАЛИЗА СТРУКТУРНОЙ ИНФОРМАЦИИ
(Вычислительные системы)
1984 год Выпуск 103

УДК 519.1

ОБ ОДНОМ АЛГОРИТМЕ КАНОНИЗАЦИИ И ОПРЕДЕЛЕНИЯ ОРБИТ ГРАФОВ

С.Х.Карабунарлиев, О.Г.Мекенян, А.А.Добрынин

Многие задачи обработки, поиска и хранения структурной информации, представляемой графами, требуют оптимальных алгоритмов канонизации. Проблема канонизации графов эквивалентна задаче распознавания изоморфизма графов и ряду других задач на графах [1,2]. Хотя ни одна из задач этого класса не кажется безусловно экспоненциальной с точки зрения временной сложности, известных полиномиальных алгоритмов в общем случае нет. Предлагаемый алгоритм предназначен для формирования канонической линейной записи графа с помеченными вершинами и ребрами, а также для определения орбит группы автоморфизмов графа. Алгоритм основан на переборной схеме с некоторыми сокращениями перебора.

Пусть $G(V,E)$ - конечный неориентированный граф порядка p без петель, где V - множество вершин графа, $|V| = p$, E - множество ребер графа. Будем называть граф помеченным, если каждому элементу множеств V и E соответствует некоторая символьная цепочка фиксированной длины. Канонической линейной записью графа назовем последовательность символов, которая однозначно задает граф независимо от исходной нумерации. Алгоритм состоит из двух частей. Первая из них является процедурой, осуществляющей предварительное разбиение множества вершин графа. Вторая часть является переборной процедурой канонизации и определения орбит группы автоморфизмов графа, использующей результаты работы первой процедуры.

1. Процедура предварительного разбиения

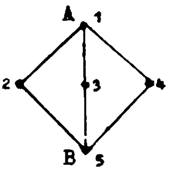
Псевдоклассами эквивалентности графа назовем любое упорядоченное разбиение множества вершин графа V на непустые подмножества V_i, $i = \overline{1,n}$, $n \le p$, инвариантное относительно автоморфизмов графа. Порядковый номер i назовем рангом псевдокласса V_i или рангом вершин из V_i. Процедура осуществляет итеративное измельчение некоторого начального разбиения вершин на псевдоклассы. Критерием остановки работы процедуры является стабилизация разбиения, т.е. совпадение вновь полученного разбиения вершин с предыдущим. Аналогичные процедуры рассматриваются в [I,3]. Шаг итеративной процедуры заключается в следующем:

I. Формировании ключей для каждой вершины. Ключом является упорядоченная последовательность рангов вершин, смежных с рассматриваемой.

2. Разбиении вершин внутри каждого нетривиального псевдокласса путем сравнения ключей.

Для начального разбиения используются метрические характеристики графа [4]. Кодом расстояния вершины v в графе назовем соответствующую вершине v строку в матрице слоев графа [5], i-я компонента которой есть мощность множества $\{u \in V \mid d(u,v)=i\}$, $i=\overline{1,e(v)}$, где $d(u,v)$ – расстояние между вершинами u и v, $e(v)$ – эксцентриситет вершины v. Начальное разбиение множества V осуществляется по меткам и кодам расстояния вершин. На рис.I приведен граф G с соответствующей ему матрицей слоев Δ и разбиением вершин на псевдоклассы.

G:



$$\Delta = \begin{Vmatrix} 3 & 1 \\ 2 & 2 \\ 2 & 2 \\ 2 & 2 \\ 3 & 1 \end{Vmatrix}$$

(1)(2 3 4)(5)

Рис. I

2. Переборная процедура

Будем считать, что граф связный. Для случая несвязного графа можно легко сделать обобщение. Основным элементом переборной процедуры является поиск в глубину с возвратом. В процессе поиска, находясь в некоторой вершине v, выбираем вершину u, смежную с v. Если вершина u уже пройдена, то выбирается другая, смежная с v вершина. Если же вершина u не пройдена, то заходим в нее и применяем процесс рекурсивно к u. Если все вершины, смежные с v, уже просмотрены, идем назад по ребру, по которому пришли в v, и продолжаем обход. Таким образом, поиск в глубину задает на каждом ребре направление прохождения и фиксирует некоторый порядок прохождения всех вершин и ребер. Исходному графу можно сопоставить ориентированный граф, соответствующий обходу графа. Ребра этого графа, приводящие в новые вершины, образуют ориентированное остовное дерево. Ребра, не принадлежащие дереву, назовем обратными ребрами. Процедура генерирует часть множества всевозможных графов прохождения исходного графа путем поиска в глубину. В ходе поиска используется информация о разбиении множества вершин на псевдоклассы процедурой предварительного разбиения:

1. Корнем дерева является вершина из псевдокласса ранга 1.

2. Из некоторой вершины заход осуществляется только в смежную вершину наименьшего ранга из всех еще не пройденных вершин.

3. После захода в новую вершину просмотр всех исходящих обратных ребер определяется порядком просмотра инцидентных им вершин.

Подобно алгоритмам [6,7] на каждом шаге поиска при заходе в некоторую вершину v по ребру (u,v) определяется часть записи, которая описывает вершину v, ребро (u,v) и обратные ребра, исходящие из v (для вершин и ребер учитываются метки и для ребер учитываются ранги вершин, инцидентных им). Ветвь перебора включается или отбрасывается путем сравнения этой части записи с соответствующей подпоследовательностью лучшей (например, лексикографически максимальной) найденной записи. Орбиты группы автоморфизмов получаются путем пошагового объединения непересекающихся множеств. В начальном состоянии каждая вершина образует отдельное подмножество. Подстановки группы автоморфизмов задаются последовательностями прохождения вершин для одинаковых линейных записей. Когда заканчивается обход графа и полученная линейная запись совпадает с лучшей из уже найденных, происходит объединение подмножеств, к

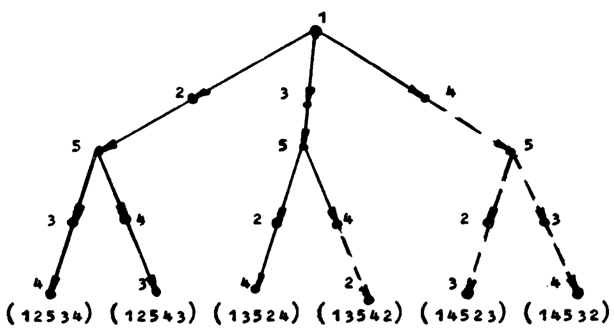


Рис. 2

которым принадлежат вершины с одинаковыми позициями в последова - тельностях прохождения вершин. Особенностью алгоритма является учет симметрии графа для сокращения перебора в отличие, например, от [8].

На рис.2 представлено дерево всех обходов графа G (рис.I). В этом дереве путь от корня дерева до висячей вершины соответствует некоторому обходу вершин графа. Порядок обхода указан около каж - дой висячей вершины (соответствующие линейные записи, порождаемые при обходах графа здесь все совпадают). Из анализа первых трех последовательностей обхода вершин графа получаем, что вершины 2,3 и 4 эквивалентны. Эта частичная информация о симметрии графа ис - пользуется для сокращения перебора. В дереве обходов на рис.2 поддеревья, подвешенные на вершине 4, можно не рассматривать в силу эквивалентности вершины 4 уже исследованным вершинам 2 и 3.

Алгоритм реализован на языке ФОРТРАН и испытывался на ЭВМ ЕС-1050.

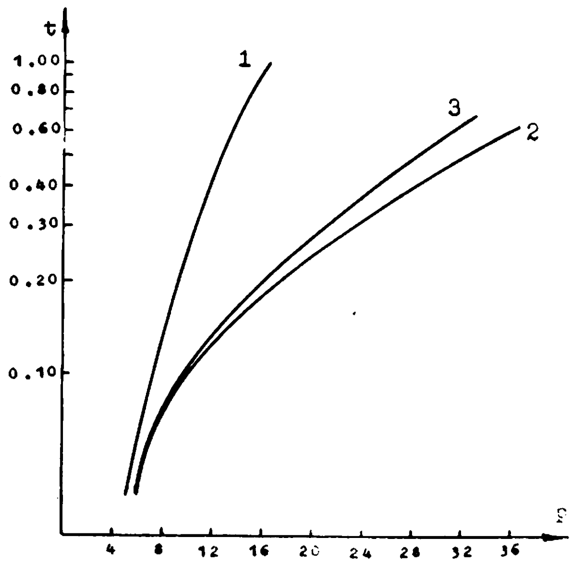


Рис. 3

На рис.3 показана зависимость временной трудоемкости t (сек) от порядка графа p для трех классов графов. Первый класс включа -

ет полные графы (кривая I), второй - жесткие кубические графы (кривая 2), третий класс включает графы, соответствующие химиче - ским соединениям, имеющим от одного до десяти циклов (кривая 3). Средняя временная трудоемкость для 30 примеров помечённых графов порядка от I5 до I7 вершин составила 0,I3 сек. Результаты счета примеров позволяют сравнить предложенный алгоритм с рядом извест- ных алгоритмов [9].

Литература

I. READ R.C., CORNEIL D.G. The Graph Isomorphism Disease. - Journal of Graph Theory,1977,v.1,p.339-363.

2. ГЭРИ М., ДЖОНСОН Д. Вычислительные машины и труднорешае - мые задачи. -М.: Мир, 1982. - 367 с.

3. CORNEIL D.G., GOTLIEB C.C. An Efficient Algorithm for Graph Isomorphism.-Journal ACM,1970,v.17,p.51-64.

4. СКОРОБОГАТОВ В.А., ХВОРОСТОВ П.В. Анализ метрических свойств графов. -В кн.: Использование ЭВМ при распознавании обра- зов (Вычислительные системы, вып. 9I). Новосибирск, 1982,с.I-2I.

5. СКОРОБОГАТОВ В.А. Относительные разбиения и слои графов. -В кн.: Вопросы обработки информации при проектировании систем (Вычислительные системы, вып. 69). Новосибирск, I977, с. 3-I0.

6. READ R.C. A New System for the Designation of Chemical Compounds for the Purposes of Data Retrieval. II.Cyclic Compounds. - Research Report CORR 80-7, University of Waterloo, Ontario,Ca - nada, 1980.

7. MORGAN M.L. The Generation of a Unique Machine Descrip - tion for Chemical Structures - a Technique Developed at CAS.- J. Chem.Documentation,1965,v.5,p.107-113.

8. McKAY B.D. Computing Algorithms and Canonical labelling of Graphs.-Lect,Notes Math.,1977,v.686,p.223-232.

9. ДОБРЫНИН А.А. Сравнение программ обработки структурной ин- формации. -Настоящий сборник, с. 90-99.

Поступила в ред.-изд.отд.
26 октября I984 года