

УДК 519.1

ЭФФЕКТИВНЫЙ АЛГОРИТМ ГЕНЕРАЦИИ ГРАФОВ
НЕРАЗВЕТВЛЕННЫХ ГЕКСАГОНАЛЬНЫХ СИСТЕМ

А.А. Добрынин

Конструктивное перечисление графов является широко применяемым средством изучения и классификации семейств графов, нахождения графов с заданными свойствами, построения контрпримеров [1]. Со стороны приложений большой интерес представляет перечисление графов, моделирующих структурные объекты в естественных науках - химии, биологии и т.п. Один из таких классов образует множество графов, соответствующих бензоидным (ароматическим) полициклическим углеводородам в органической химии. Графы этого класса можно представить в виде соединенных друг с другом определенным образом шестиугольников на плоскости. Библиография работ, посвященных перечислению различных подклассов графов бензоидных углеводородов и описанию алгоритмов их генерации, насчитывает десятки наименований ([2-17] - список выборочный). Результаты перечисления графов, полученные до середины 1987 года, даны в обзоре [2]. В настоящей работе предложен новый алгоритм конструктивного перечисления графов неразветвленных ката-конденсированных бензоидных углеводородов.

§1. Основные определения

Рассматриваются конечные неориентированные связные графы без петель и кратных ребер, $V(G)$ - множество вершин графа G ,

$|V(G)| = n_G$ - порядок графа. Определим класс графов, в которых все внутренние грани являются шестиугольными, и две произвольные грани либо являются смежными, т.е. имеют единственное общее ребро, либо не имеют общих вершин. Поставив в соответствие каждой шестиугольной грани новую вершину и соединив такие вершины, если соответствующие грани являются смежными, получим характеристический граф исходного графа. Обозначим K_6 класс графов, для которых характеристический граф изоморфен простой цепи. Графы класса K_6 моделируют молекулярные структуры неразветвленных (характеристический граф не содержит вершин степени более двух) ката-конденсированных (характеристический граф не содержит циклов) бензоидных углеводородов. На рис.1 графы G_1 , G_2 и G_3 принадлежат классу K_6 . Ясно, что порядок графа из K_6 ,

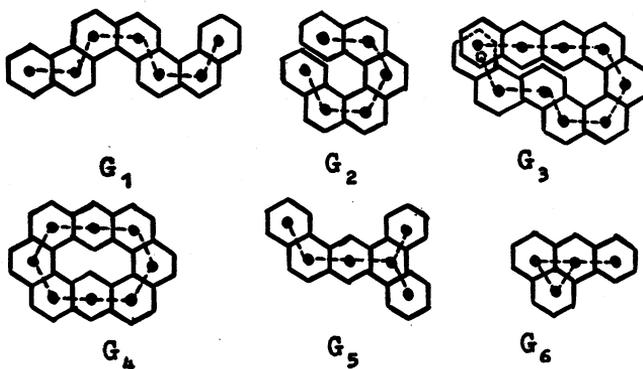


Рис. 1

состоящего из n колец, равен $4n + 2$ и все вершины графа имеют степень 2 или 3. Шестиугольные грани вместе с их границей назовем кольцами графа. Концевым кольцам соответствуют вершины характеристического графа степени один. Все графы класса K_6 можно разбить на два непересекающихся подмножества, одно из которых содержит графы, вложимые в правильную гексагональную

решетку на плоскости, а другое содержит графы, не вложимые в решетку. Определим границу графа как подграф, содержащий все ребра, каждое из которых принадлежит только единственному кольцу. Если при расположении графа на плоскости все внутренние грани являются шестиугольными, то все ребра границы графа будут смежны с бесконечной гранью. Полуграница графа есть связный подграф границы, содержащий по два ребра каждого концев



Рис. 2

кольца. Для графа G на рис.2 показаны его граница и полуграница. Остальная терминология и обозначения соответствуют [18].

§2. Построение алгоритмов генерации

Рассмотрим схему построения группы алгоритмов генерации графов, моделирующих бензоидные углеводороды. В силу особенностей структуры графов в алгоритмах генерации графы представляются в виде слов (цепочек символов) над некоторым алфавитом A . Элементы алфавита описывают возможное расположение вершин, ребер или граней графов на плоскости (в правильной гексагональной решетке) или их другие характеристики, а слово, соответствующее графу, формируется при обходе его вершин, ребер или граней. Для представления графов применялись степенные (вершинные) и реберные граничные последовательности [4,6,8,10,12,13], периметрический код [11], код характеристического графа [9,14] и другие [7]. На множестве слов, определяющих один и тот же граф, вследствие произвольности начала обхода элементов графа, определяется канонический вид. Алгоритмы перечисления, как правило, включают следующие этапы:

1) организуется процесс порождения слов заданной длины над алфавитом A . При этом не обязательно генерируются все слова при условии, что для каждого графа хотя бы одно соответствующее ему слово не будет пропущено.

Для каждого порождаемого слова проверяется:

2) соответствует ли полученное слово некоторому графу из рассматриваемого класса (допустимость слова);

3) если слово является допустимым, то не был ли на предыдущих шагах алгоритма получен граф, изоморфный графу, определяемому этим словом.

Трудоемкость алгоритма генерации графов в большой степени определяется эффективностью построения указанных этапов. Длина l слова, представляющего граф, во многих алгоритмах заключается в пределах $h - 2 \leq l \leq p$, где h - число шестиугольных колец графа (порядок характеристического графа), p - порядок графа, а мощность алфавита лежит в границах $2 \leq |A| \leq 6$. Так, для представления графов в [19] используются слова длины $l = h - 2$ и алфавит мощности $|A| = 3$. Чем больше длина слов и мощность алфавита, тем большее количество слов необходимо проверять на их допустимость или больше осуществлять проверок ограничений при построении допустимого слова. Сравнение числа обрабатываемых слов с числом графов дает первую информацию об эффективности алгоритма. Для проверки допустимости слова формулируются необходимые и достаточные условия, гарантирующие принадлежность соответствующего графа заданному классу. Типичными условиями являются наличие (отсутствие) вхождений заданных подслов или выполнение соотношений для элементов подслов исходного слова [14]. Если достаточных условий для проверки допустимости слова сформулировать не удастся, применяется другой способ порождения слов. Пусть имеется множество слов, представляющих все попарно неизоморфные графы заданного класса с числом шестиугольных колец $h - 1$. Это множество слов расширяют та-

ким образом, чтобы вновь построенное множество включало бы только допустимые слова для графов с числом колец h . Условия для изменения слова (присоединения одного нового кольца к графу) являются обычно простыми и не требуют перебора подслов. В результате из каждого слова образуется множество допустимых слов большей длины, некоторые из них могут совпадать со словами, полученными при увеличении других слов.

Наиболее неэффективной процедурой установления изоморфности полученного на некотором шаге генерации графа является проверка на совпадение его слова (в канонической форме) со словами всех графов, полученных на предыдущих шагах. Наиболее эффективная процедура устанавливает повторность получаемого на текущем шаге генерации графа за константное число операций. Таким образом, алгоритм конструктивного перечисления указанного вида требует не менее $CN(m+k)$ операций, где N - число порождаемых слов заданной длины, k - число операций для определения допустимости слова для представления графа, m - число операций определения изоморфности графа ранее полученным графам, C - константа. В алгоритме генерации, рассматриваемом в настоящей работе, величины m и k являются константами, граф представляется словом длины $l = h - 2$ и $|A| = 2$.

§3. Представление полуграниц графа

Здесь и далее рассматриваются графы K_6 . Под обходом полуграницы графа будем понимать последовательный просмотр ее ребер и вершин, начиная с концевого кольца. Два кольца графа называются парными, если они не являются смежными в графе, но имеют смежные вершины. Если эти вершины принадлежат полугранице $G_1 \subseteq G$, то парные кольца назовем парными на полугранице G_1 . В графе на рис. 3а кольца 2 и 4 являются парными на полугранице G_1 , а кольца 3, 5 и 4, 6 являются парными на полугранице G_2 . Поставим в соответствие графу G пару графов специального ви-

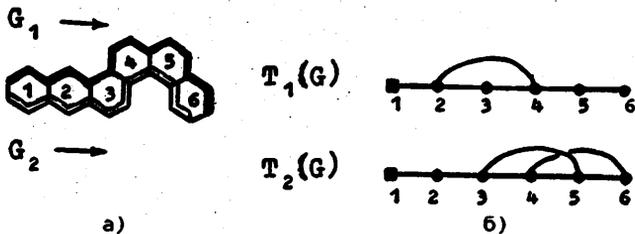


Рис. 3

да, каждый из которых отражает парность колец при обходе соответствующей полуграницы. Обход обеих полуграниц начинается с одного и того же концевго кольца. Соединим в характеристическом графе ребром каждую пару вершин, соответствующих парным кольцам на одной из полуграниц. Построенный граф назовем графом соответствующей полуграницы. Таким образом, полуграницы G_1 и G_2 графа G на рис.3а представляются графами полуграниц $T_1(G)$ и $T_2(G)$ (рис.3б). Вершина характеристического графа, соответствующая концевому кольцу, с которого начинается обход полуграниц, будет изображаться левее остальных вершин в виде квадрата. Ребра, изображенные отрезками прямых, показывают смежность колец графа, а "криволинейные" ребра отражают парность колец при обходе полуграниц. "Криволинейные" ребра будем называть дугами. По определению парных колец дугами в графах полуграниц могут быть соединены вершины, находящиеся друг от друга в характеристическом графе на расстоянии, равном 2. Так как характеристический граф изоморфен простой цепи, то в графе полуграницы одна вершина может быть соединена дугами не более чем с двумя другими вершинами. На рис. 4 приводятся графы с числом колец $h = 5$ и их графы полуграниц. Из построения графов полуграниц следует

УТВЕРЖДЕНИЕ 1. Для любой вершинки v в графах полуграниц выполняется $2 \leq \text{deg } v \leq 4$, если v есть не кон-

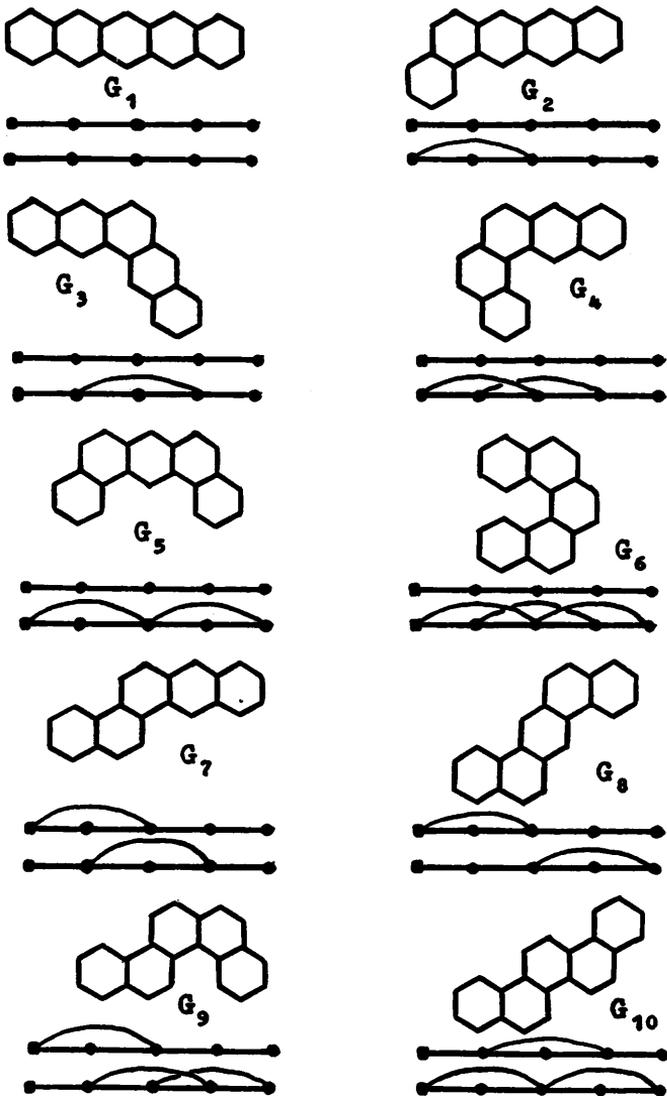


Рис. 4. Графы полуграниц для графов из K_6 с $h = 5$

цевая вершина характеристического графа, $2 \leq \deg v \leq 3$, если v смежна с концевой вершиной, и $1 \leq \deg v \leq 2$ для концевой вершины, где $\deg v$ - степень вершины v .

Особенности рассматриваемого класса графов налагают ограничения на структуру графов полуграниц. Следующее свойство является основным для графов полуграниц T_1 и T_2 .

УТВЕРЖДЕНИЕ 2. Если вершины соединены дугой в T_1 (или T_2), то они не соединены дугой в T_2 (или T_1).

Доказательство следует из того, что если некоторая пара вершин соединена дугой в обоих графах полуграниц, то характеристический граф исходного графа должен иметь вершину степени более двух, чего не может быть для графов рассматриваемого класса.

Из утверждения 2 следует, что если для графов полуграниц T_1 и T_2 выполняется $T_1 \cong T_2$, то существует единственный граф с таким свойством для заданного числа колец (граф G_1 на рис. 4), а графы полуграниц изоморфны простой цепи. Пусть T_1 и T_2 - произвольные графы одинакового порядка. Условия соответствия T_1 и T_2 графам полуграниц некоторого графа устанавливает

УТВЕРЖДЕНИЕ 3. Если графы T_1 , T_2 представимы в описанном выше виде и выполняются утверждения 1-2, то существует единственный граф, для которого T_1 и T_2 являются графами его полуграниц.

Для доказательства заметим, что по графам T_1 и T_2 , удовлетворяющих условиям утверждения 3, однозначно восстанавливается последовательность присоединения колец к начальному графу, состоящему из пары колец, что однозначно определяет и смежность вершин графа.

Приведем некоторые свойства графов полуграниц. Пусть T_1 - произвольный граф порядка n , удовлетворяющий утверждениям 1-2. Тогда

- 1) если граф T_2 изоморфен простой цепи порядка h , то T_1 и T_2 являются графами полуграниц некоторого графа;
- 2) если вершины T_1 имеют максимально возможную степень, то существует единственная пара графов T_1 и T_2 , которые являются графами полуграниц некоторого графа, граф T_2 изоморфен простой цепи порядка h ;
- 3) если в графе T_1 существуют m пар вершин, которые можно соединить дугой, не нарушая утверждений 1-2, то существует 2^m графов T_2 таких, что T_1 и T_2 образуют графы полуграниц некоторого графа.

§4. Свойства графов полуграниц

Сформулируем в терминах графов полуграниц условия изоморфности пары графов из K_6 и условия симметричности графа. Если графы G и H являются изоморфными, то при отображении G и H друг на друга их графы полуграниц должны также переходить друг в друга. Так как каждую полуграницу можно обходить в двух разных направлениях и граф имеет две полуграницы, то для определения изоморфности пары графов необходимо и достаточно проверить совпадение графов их полуграниц. Для графа полуграницы T определим операцию инверсии, результатом которой является граф полуграницы T^* , получаемый при обходе той же полуграницы, но с противоположного концевое кольца (рис.5). Ясно, что $(T^*)^* \cong T$ и пары графов полуграниц (T_1, T_2) и (T_1^*, T_2^*) определяют один и тот же граф. В следующем утверждении учитываются возможные способы совпадения полуграниц графов.

УТВЕРЖДЕНИЕ 4. Графы G и H изоморфны, если и только если выполняется одно из условий:

- 1) $T_1(G) \cong T_1(H)$ и $T_2(G) \cong T_2(H)$;
- 2) $T_1(G) \cong T_2(H)$ и $T_2(G) \cong T_1(H)$.

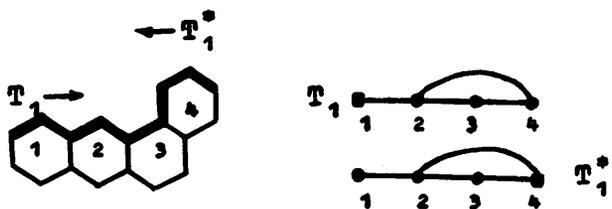


Рис.5. Операция инверсии для графа полуграницы

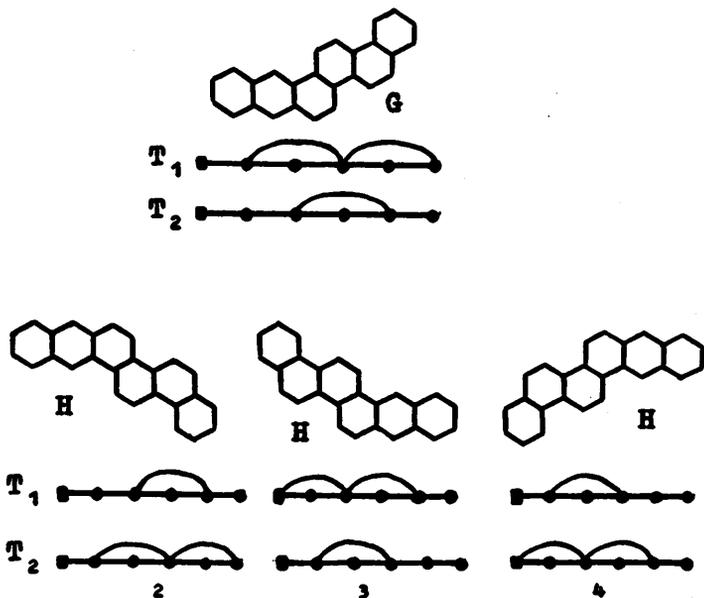


Рис.6. Совпадение графов полуграниц при изоморфизме

$$3) T_1(G) \cong T_1^*(H) \quad \text{и} \quad T_2(G) \cong T_2^*(H);$$

$$4) T_1(G) \cong T_2^*(H) \quad \text{и} \quad T_2(G) \cong T_1^*(H).$$

На рис. 6 показаны различные случаи совпадения графов полуграниц для пары графов. Назовем граф симметричным, если его группа автоморфизмов не является тождественной. При некотором автоморфизме графа (изоморфизме на себя) полуграницы графа переходят друг в друга, Таким образом, утверждение 4 можно переформулировать для распознавания типа симметрии графа.

УТВЕРЖДЕНИЕ 5. *Граф G с числом колец h является симметричным, если и только если выполняется одно из условий:*

1) $T_1(G) \cong T_2(G)$. *Граф G не имеет парных колец (граф единственный);*

2) $T_1(G) \cong T_1^*(G)$ и $T_2(G) \cong T_2^*(G)$. *Граф G является зеркально-симметричным; при отображении графа на себя вершины i -го кольца отображаются на вершины $(h-i+1)$ -го кольца одной и той же полуграницы, и обратно (нумерация колец последовательная);*

3) $T_1(G) \cong T_2^*(G)$. *Граф G является центрально-симметричным; при отображении графа на себя вершины i -го кольца одной полуграницы отображаются на вершины $(h-i+1)$ -го кольца другой, и обратно.*

Все виды симметрии графов и их полуграницы представлены на рис. 4; случаю 1 соответствует граф G_1 , графы G_1, G_3, G_5, G_6 и G_{10} являются зеркально-симметричными, а графы G_1 и G_8 - центрально-симметричными.

§5. Представление графов полуграниц

Переход от исходного графа из K_6 к графам полуграниц, имеющих также сложную структуру, объясняется тем, что последние можно представить двоичными наборами с сохранением всех свойств

графов полуграниц. Пусть граф G состоит из h колец, а вершины графа полуграницы $T(G)$ последовательно пронумерованы по направлению обхода полуграницы. Графу $T(G)$ поставим в соответствие двоичный набор n из h элементов (разрядов), в котором i -й элемент n_i , $i = 1, 2, \dots, h$, определяется как

$$n_i = \begin{cases} 1, & \text{если вершины } i \text{ и } i+2 \text{ соединены в } T(G) \text{ дугой;} \\ 0 & \text{в противном случае.} \end{cases}$$

По построению двоичного набора n два последних разряда будут равны нулю для любых графов полуграниц, так что их можно отбросить. Легко показать, что соответствие графов полуграниц и их двоичных представлений является взаимно-однозначным. Следовательно, граф из h колец представляется в виде двух двоичных наборов $n_1(G)$ и $n_2(G)$ разрядности $h-2$, $G = (n_1(G), n_2(G))$. Двоичные наборы будем называть кодами полуграниц графа или просто кодами графа. Графы с числом колец $h = 5$ на рис. 4 имеют коды $G_1 = (000,000)$, $G_2 = (000,100)$, $G_3 = (000,010)$, $G_4 = (000,110)$, $G_5 = (000,101)$, $G_6 = (000,111)$, $G_7 = (100,010)$, $G_8 = (100,001)$, $G_9 = (100,011)$, $G_{10} = (010,101)$. Аналогично утверждения 2-3 для двоичных наборов сформулируем

УТВЕРЖДЕНИЕ 6. *Два двоичных набора n_1 и n_2 разрядности $h-2$ определяют некоторый граф, если и только если поразрядная логическая операция "И" над наборами равна нулю, т.е. $n_1 \wedge n_2 = 0$.*

Распространим операцию инверсии графов полуграниц на их двоичные коды. Так как для графа полуграницы $T(G)$ из $h+2$ вершин граф $T^*(G)$ определяется обратным обходом полуграницы и отношение парности колец является симметричным, то $n^*(G)$ есть двоичный набор, в котором i -й разряд совпадает с $(h-i+1)$ -м разрядом двоичного набора $n(G)$ для $i = 1, 2, \dots, h$. Таким образом, $n^*(G)$ получается обратной перенумерацией разрядов $n(G)$. Поэтому утверждения 4-5 об установлении

изоморфизма графов и определении типа симметрии графа без изменений переносятся на двоичные коды полуграниц. Коды всех графов рассматриваемого класса содержатся в множестве $M \subseteq \mathbb{N} \times \mathbb{N}$, $M = \{(n_1, n_2), n_1 \wedge n_2 = 0\}$. Отображение $f: K_6 \rightarrow M$ не является однозначным, так как одному и тому же графу могут соответствовать различные коды, например, если $(n_1, n_2) \in M$, то и $(n_1^*, n_2^*) \in M$.

§6. Алгоритм генерации графов

Коды полуграниц графа (двоичные наборы) удобно рассматривать как числа. На множестве кодов введем естественный порядок и считаем, что для графа G с кодами $n_1(G)$ и $n_2(G)$ всегда выполняется $n_1(G) < n_2(G)$. Равенство $n_1(G) = n_2(G)$ выполняется для единственного графа без парных колец. Алгоритм генерации графов состоит в генерации пар чисел, соответствующих графам класса K_6 , и проверке полученного на некотором шаге алгоритма графа на новизну. Обозначения в алгоритме: N - количество чисел разрядности $h - 2$, n_1 и n_2 - числа-кандидаты на коды полуграниц графа.

Алгоритм 1. Генерация графов класса K_6

Вход: h - число колец в порождаемых графах.

Выход: COUNT - число всех неизоморфных графов из h колец.

Метод: $N + 2^{h-1} - 1$;

COUNT + 1;

для $n_1 + 0$ до N шаг 1 цикл;

для $n_1 + n_1 + 1$ до N шаг 1 цикл;

если $n_1 \wedge n_2 = 0$ то

если $\left\{ \begin{array}{l} \text{ранее не было получено} \\ \text{графа, определяемого парой } (n_1, n_2) \end{array} \right\}$

то COUNT ← COUNT + 1;

все;

все;

Конец;

УТВЕРЖДЕНИЕ 7. Проверка на изоморфность получаемого в алгоритме 1 графа производится за число операций, ограниченное константой.

Действительно, пусть на текущем шаге алгоритма 1 получен граф G с кодами n_1 и n_2 , $n_1 < n_2$. Графу G изоморфен граф с кодами n_1^* и n_2^* , причем порядок на числах n_1^* , n_2^* заранее не известен. Необходимо установить, были ли значения n_1^* и n_2^* в качестве кодов графа просмотрены на предыдущих шагах алгоритма 1. Если это так, то текущие коды графа n_1 и n_2 отбрасываются. Для значений n_1^* и n_2^* выполняется одно из неравенств $n_1^* < n_2^*$ или $n_2^* < n_1^*$. Общая схема условий для отбрасывания изоморфных графов при $n_1^* < n_2^*$ имеет вид:

$$n_1^* < n_2^* \rightarrow \begin{cases} n_1 > n_1^* - \text{граф } G \text{ получен ранее;} \\ n_1 < n_1^* - \text{граф } G \text{ является новым;} \\ n_1 = n_1^* \rightarrow \begin{cases} n_2 > n_2^* - \text{граф } G \text{ получен ранее;} \\ n_2 < n_2^* - \text{граф } G \text{ является новым.} \end{cases} \end{cases}$$

Для $n_1^* > n_2^*$ в вышеприведенной схеме значения n_1^* и n_2^* нужно поменять местами. Следовательно, для определения неизоморфности графа, полученного на текущем шаге алгоритма, графам, полученным на предыдущих шагах алгоритма, достаточно четырех операций сравнения кодов графа. Фактически в алгоритме генерации перечисляется множество значений взаимнооднозначного отображения $f: K_6 \rightarrow M$, где $M \subseteq N \times N$ и $M = \{(m_1, m_2) \mid (m_1, m_2) = \min[(n_1, n_2), (k_1, k_2)], n_1 \leq n_2, n_1 \wedge n_2 = 0, k_1 = \min\{n_1^*, n_2^*\}, k_2 = \max\{n_1^*, n_2^*\}\}$, где порядок на двойках чисел определяется покомпонентным сравнением. Полная схема генерации графов представлена алгоритмом 2.

Алгоритм 2. Генерация графов класса K_6

Вход: h - число колец в порождаемых графах.

Выход: COUNT - число неизоморфных графов из h колец.

Метод: $N \leftarrow 2^{h-1} - 1;$
COUNT $\leftarrow 1;$

для $n_1 \leftarrow 0$ до N шаг 1 цикл;

для $n_2 \leftarrow n_1 + 1$ до N шаг 1 цикл;

если $n_1 \wedge n_2 \neq 0$ то идти на КОН;

если $n_1^* < n_2^*$ то

$$\left\{ \begin{array}{l} \text{если } n_1 > n_1^* \text{ то идти на ИЗМ;} \\ \text{если } n_1 < n_1^* \text{ то идти на НОВ;} \\ \text{если } n_2 > n_2^* \text{ то идти на ИЗМ;} \end{array} \right\}$$

иначе

$$\left\{ \begin{array}{l} \text{если } n_1 > n_2^* \text{ то идти на ИЗМ;} \\ \text{если } n_1 < n_2^* \text{ то идти на НОВ;} \\ \text{если } n_2 > n_1^* \text{ то идти на ИЗМ;} \end{array} \right\}$$

НОВ: COUNT \leftarrow COUNT + 1;

ИЗМ: {граф с кодом (n_1, n_2) изоморфен полученному ранее}

КОН: {наборы n_1 и n_2 не определяют граф}

все $n_2;$

все $n_1;$

Конец;

Количество графов $C(h)$, состоящих из h колец, определяется формулой [5]:

$$C(h) = \begin{cases} \frac{1}{4} (3^{(h-2)/2} + 1)^2 & \text{для } h = 2, 4, 6, \dots; \\ \frac{1}{4} (3^{h-2} + 3^{(h-1)/2} + 3^{(h-3)/2} + 1) & \text{для } h = 3, 5, 7, \dots \end{cases}$$

Трудоёмкость алгоритма конструктивного перечисления графов этого класса, затрачивающего на получение каждого графа не

более k операций, не менее $kC(h)$ операций или $O(3^h)$ при k , не зависящем от h .

УТВЕРЖДЕНИЕ 8. Трудоемкость алгоритма 2 есть $O(4^h)$.

Для получения кода графа и проверки графа на новизну требуется не более 10 операций, а в двойном цикле просматривается не более 4^{h-1} пар чисел-кандидатов на код графа. Инверсные наборы n^* можно вычислить до выполнения двойного цикла и расположить их в массиве A таком, что $A(n) = n^*$. Трудоемкость вычисления инверсных наборов (заполнение массива A) составляет $O(h \cdot 2^h)$ операций, следовательно, трудоемкость алгоритма 2 в целом есть $O(4)^h$.

В алгоритме возможны дальнейшие улучшения, позволяющие уменьшить константы в оценках трудоемкости. Например, при нечетных значениях n_1 нечетные значения n_2 можно не рассматривать, так как всегда выполняется $n_1 \wedge n_2 \neq 0$.

Алгоритм 2 реализован на языке Фортран-4 в ОС RT-11. Коды полуграниц представлялись целыми числами, при формировании наборов n^* использовались логические операции. В левой части таблицы приводятся количество графов класса K_6 в семействах с различным числом колец и время генерации графов (ЭВМ "Электроника-60", процессор типа M2).

§7. Система кодирования графов класса K_6

Представление графов в виде кодов полуграниц является удобным внутримашинным представлением, предназначенным для генерации, обработки и хранения данных о графах, однако оно достаточно трудоемко для получения кода графа человеком, например, при вводе графов с дисплея по их изображениям. Для этих целей можно использовать "внешнее" представление графов класса K_6 . Пусть кольца графа, состоящего из h колец, пронумерованы последовательно. В новом представлении графа указывается число колец графа (если необходимо) и две последовательности

Т а б л и ц а

Время генерации графов с различным числом колец

Число колец n	Графы класса K_6		Графы, вложимые в гексагональную решетку на плоскости	
	Число графов	Время генерации	Число графов	Время генерации
1	1	-	1	-
2	1	-	1	-
3	2	-	2	-
4	4	-	4	-
5	10	-	10	-
6	25	-	24	-
7	70	-	67	1 с
8	196	-	182	3 с
9	574	1 с	520	11 с
10	1.681	3 с	1.474	36 с
11	5.002	11 с	4.248	1 мин 51 с
12	14.884	42 с	12.196	5 мин 51 с
13	44.530	2 мин 43 с	35.168	18 мин 43 с
14	133.225	10 мин 37 с	101.226	1 ч 0 мин 34 с
15	399.310	41 мин 47 с	291.565	-
16	1.196.836	2 ч 45 мин 4 с	838.764	-

чисел, соответствующие полуграницам и содержащие номера парных колец графа. На рис. 7 показаны примеры задания графов. Если

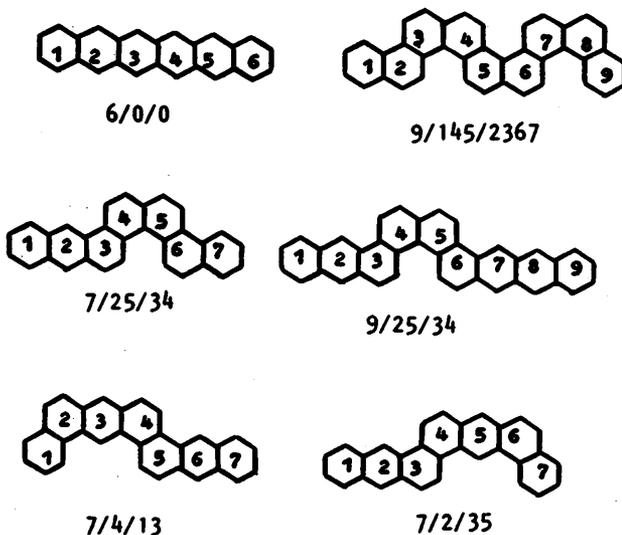


Рис. 7

парных колец на полугранице нет, то указывается ноль. Это представление можно получить и из кодов полуграниц, указывая вместо единицы номер соответствующего кольца и не учитывая нули. Поэтому для новых кодов сохраняются свойства кодов полуграниц: номер кольца может присутствовать в коде только один раз; максимальное число номеров колец в коде не более $h - 2$; коды $h/i_1, \dots, i_n/j_1, \dots, j_m/$ и $h/j_1, \dots, j_m/i_1, \dots, i_n$, $h/i_1, \dots, i_n/j_1, \dots, j_m$ и $h/h-i_n-1, \dots, h-i_1-1/h-j_m-1, \dots, h-j_1-1$ определяют один и тот же граф; по кодировке определяется вид симметрии графа и т.п. Графы семейства на рис. 4 с $h = 5$ можно задать как 0/0, 0/1, 0/2, 0/12, 0/13, 0/123, 1/2, 1/3, 1/23, 2/13. Предложенная система кодирования ясна для понимания и не требует больших затрат для получения кода.

§8. Алгоритмы обработки структурной информации

Представление графов класса K_6 в виде кодов полуграниц можно использовать при построении различных алгоритмов обработки структурной информации. Ниже рассматриваются алгоритмы восстановления матрицы смежности графа по его коду, проверки графа на вложимость в правильную гексагональную решетку на плоскости, определения вложения одного графа в другой, нахождения пересечения графов (максимального общего подграфа), вычисления координат вершин и дистанции графа (индекса Винера). Оценка трудоемкости алгоритмов будет определяться в предположении, что операция сравнения или поразрядная логическая операция над двумя двоичными наборами длины n выполняется за один шаг.

Восстановление матрицы смежности. Матрица смежности графа является стандартным представлением графа во многих алгоритмах и программах обработки графов. Алгоритм восстановления матрицы смежности состоит в анализе места присоединения колец путем последовательного просмотра разрядов кодов полуграниц. Пусть для графа G_{i-1} , состоящего из $i - 1$ колец, матрица смежности уже построена. Присоединим к ребру (u, v) графа G_{i-1} новое кольцо

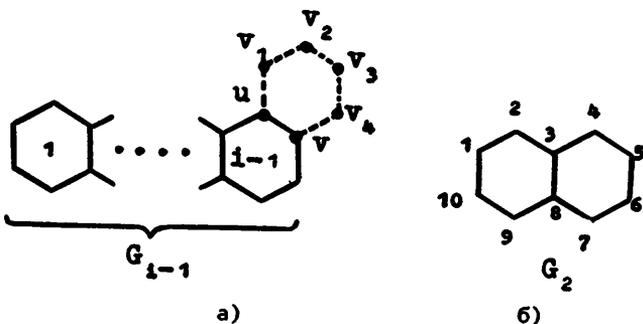


Рис. 8

цо (рис. 8а). Матрица смежности графа G_i включает матрицу смежности G_{i-1} и информацию о смежности вершин в цепи $u, v_1,$

v_2, v_3, v_4, v . Работа алгоритма начинается с присоединения кольца к графу G_2 , состоящему из двух колец. Вершины нумеруются последовательными целыми числами, начиная с 1.

Алгоритм 3. Восстановление матрицы смежности

Вход: 1. h - число колец графа G_h .

2. n_1, n_2 - коды полуграниц графа G_h .

3. Матрица MS размерности $(4h+2) \times (4h+2)$ с нулевыми элементами.

Выход: MS - матрица смежности графа G_h .

Метод: 1. Отметить в MS смежность вершин графа G_2 , состоящего из двух колец (рис. 86).

2. $N \leftarrow 10$;

3. для $i \leftarrow 1$ до $h-2$ шаг 1 цикл;

3.1. если { i -й разряд в n_1 есть 1} то
{ $u \leftarrow N-2$; $v \leftarrow N-1$ }

иначе

если { i -й разряд в n_2 есть 1} то
{ $u \leftarrow N-2$; $v \leftarrow N-1$ }

иначе

{ $u \leftarrow N-1$; $v \leftarrow N$ };

3.2. занести в матрицу MS информацию о смежности вершин цепи с номерами $u, N+1, N+2, N+3, N+4, v$; MS - матрица смежности графа G_{i+2} ;

3.3. $N \leftarrow N+4$;

все;

Конец;

Из построения алгоритма 3 следует, что его трудоемкость пропорциональна числу колец графа.

Вложение графа в гексагональную решетку. Введем на плоскости координатную систему, "точки" которой есть шестиугольные грани правильной шестиугольной решетки. Изменение координат

при переходе от одной точки к другой показано на рис.9а [13]. Алгоритм заключается в последовательном размещении колец в решетке в соответствии с содержанием кодов полуграниц. Кольцам графа из двух колец присваиваются начальные координаты. В процессе построения графа координаты, приписываемые его кольцам, помечаются как занятые. Кроме того, помечаются как занятые и все координаты колец решетки, которые смежны с кольцами графа, за исключением последнего присоединенного кольца (рис.9б). Если при присвоении координат очередному присоединяемому кольцу координаты оказываются занятыми, то исходный граф не вкладывается в гексагональную решетку. Для вычисления координат текущего присоединяемого кольца необходимо учитывать информацию о присоединении предыдущего кольца. Пронумеруем направления, по которым к кольцу может присоединяться другое кольцо (рис.10а). Пусть i -е кольцо присоединено к $(i-1)$ -му по направлению 5 и имеет координаты (x, y) (рис.10б). Фрагмент алгоритма определения координат для $(i+1)$ -го кольца имеет вид:

```

если НАПРАВЛЕНИЕ=5 то
    если  $\{(i-1)\text{-й разряд } n_1 \text{ равен } 1\}$  то
        {координаты  $(i+1)$ -го кольца  $(x-1, y+1)$ }
    иначе
        если  $\{(i-1)\text{-й разряд } n_2 \text{ равен } 1\}$  то
            {координаты  $(i+1)$ -го кольца  $(x+2, y)$ }
        иначе
            {координаты  $(i+1)$ -го кольца  $(x+1, y+1)$ }.

```

Для каждого другого значения направления присоединения i -го кольца координаты $(i+1)$ -го кольца вычисляются аналогичным образом. Очевидно, что число операций для вычисления координат одного кольца ограничено сверху константой, следовательно, алгоритм проверки на вложимость графа в правильную гексагональную решетку на плоскости имеет трудоемкость $O(n)$. Та-

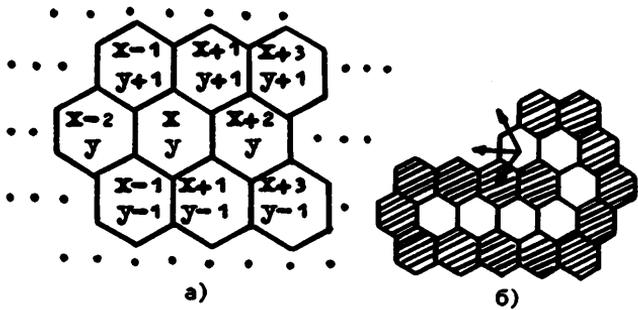


Рис.9. Координаты колец гексагональной решетки

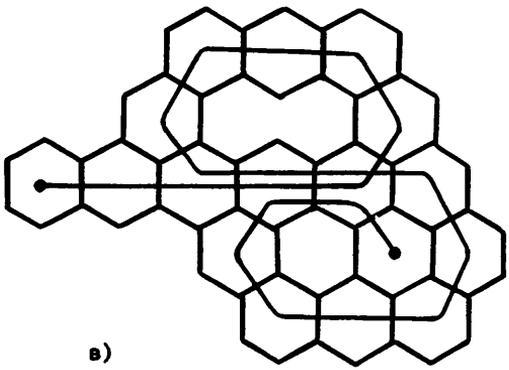
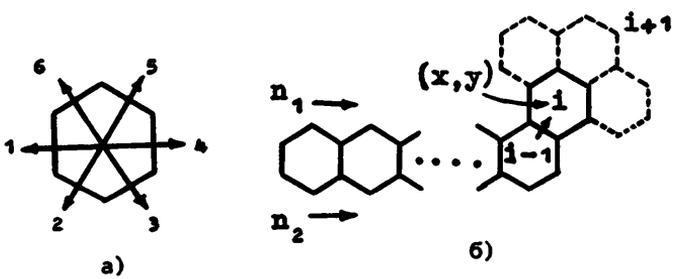


Рис.10. К проверке вложения графа в гексагональную решетку

ким образом, алгоритм конструктивного перечисления всех графов из K_6 , вкладываемых в решетку, имеет трудоемкость $O(h^4)$. В правой части таблицы приводятся количество вложимых в решетку графов для каждого значения h и времена генерации семейств графов.

Графы класса K_6 удобно изображать в гексагональной решетке, поэтому рассматриваемый алгоритм можно использовать и для визуализации графов. В зависимости от направления и содержимого кодов n_1 и n_2 вершинам вновь присоединяемого кольца приписываются координаты на плоскости в декартовой системе координат.

При расположении в решетке графа, который в нее не вкладывается, различные кольца графа располагаются на одном и том же месте, вследствие чего при визуализации и восприятии изображения граф не может быть однозначно идентифицирован. Однако если удобным образом метить кольца графа, то неопределенности можно избежать. Например, на рис.10 в гексагональной решетке размещен граф из 22 колец, которые накладываются друг на друга до трех раз. Изображение характеристического графа позволяет однозначно определять порядок следования колец графа при наложениях.

Вычисление дистанции графа (индекса Винера). Дистанция графа является метрической характеристикой графа (инвариантом) [20], определяемой выражением

$$D(G) = \frac{1}{2} \sum_{u, v \in V(G)} d(u, v),$$

где $d(u, v)$ - расстояние между вершинами u и v в графе G . Для молекулярных графов в органической химии дистанция графа известна как "число Винера" и широко используется как топологический индекс в задачах установления связи "структура-свойство" [21,22]. В [23] описываются зависимости между числом Ви-

нера графов полициклических гексагональных структур и их физико-химическими свойствами. В [19] изучалось распределение значений числа Винера для вложенных в решетку графов класса K_6 с $h \leq 9$ кольцами путем конструктивного перечисления графов и вычисления числа Винера. Вследствие экспоненциального роста числа графов при увеличении числа колец h понижением трудоемкости вычисления числа Винера увеличивается эффективность и возможности такого рода исследований. Простой алгоритм вычисления числа Винера по списку или матрице смежности графа, очевидно, имеет трудоемкость $O(h^2)$.

УТВЕРЖДЕНИЕ 9. По кодам полуграниц графа вычисление индекса Винера осуществляется с трудоемкостью $O(h)$.

Пусть граф G_h из h колец получается присоединением кольца к ребру (u, v) некоторого графа G_{h-1} . Следуя [24], выразим дистанцию графа G_h через дистанцию графа G_{h-1} и дистанции вершин u и v :

$$D(G_h) = D(G_{h-1}) + 2(D_{G_{h-1}}(u) + D_{G_{h-1}}(v)) + 2(12h-1),$$

где $D_G(v) = \sum_{u \in V(G)} d(u, v)$ - дистанция вершины v в графе G .

Решая возникающее рекуррентное уравнение

$$\begin{cases} D_h = D_{h-1} + 2\Phi_{h-1} + 2(12h-1), \\ D_1 = 27, \Phi_1 = 18, \end{cases}$$

получаем $D_h = D(G_h) = 2 \sum_{i=1}^{h-1} \Phi_i + 2(h-1)(6h+11) + 27$.

Таким образом, трудоемкость вычисления D_h определяется трудоемкостью вычисления дистанций пар вершин, входящих в Φ_i , $i = 1, 2, \dots, h-1$. Рассмотрим граф G_{h-1} на рис.11, к которому присоединяется кольцо. Различные возможности места присоеди-

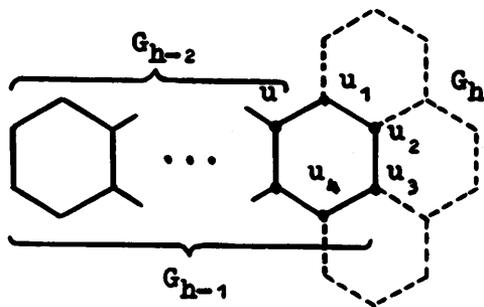


Рис. 11

ния кольца изображены пунктиром. Для вычисления D_h нужно определить дистанции вершин ребер (u_1, u_2) , (u_2, u_3) , (u_3, u_4) , к одному из которых будет присоединено кольцо. Легко подсчитать, что если кольцо присоединено к ребру (u_1, u_2) , то выполняются равенства

$$D'_{G_{h-1}}(u_1) = D_{G_{h-2}}(u) + [4(h-2) + 2] + 6,$$

$$D_{G_{h-1}}(u_2) = D_{G_{h-2}}(u) + 2[4(h-2) + 2] + 4,$$

при присоединении кольца к ребру (u_2, u_3) имеем

$$D_{G_{h-1}}(u_2) = D_{G_{h-2}}(u) + 2[4(h-2) + 2] + 4,$$

$$D_{G_{h-1}}(u_3) = D_{G_{h-2}}(v) + 2[4(h-2) + 2] + 4,$$

а если кольцо присоединения к ребру (u_3, u_4) , то

$$D_{G_{h-1}}(u_3) = D_{G_{h-2}}(v) + 2[4(h-2) + 2] + 4,$$

$$D_{G_{h-1}}(u_i) = D_{G_{h-2}}(v) + [4(h-2) + 2] + 6.$$

Отсюда следует, что вычисление Φ_i для всех $i = 1, 2, \dots, h-1$ можно организовать в виде одного итеративного процесса.

В представленном ниже алгоритме вершина u всегда находится на одной полугранице, а вершина v - на другой. К ребру (u, v) происходит присоединение кольца на очередном шаге алгоритма.

Алгоритм 4. Вычисление дистанции графа (индекса Винера)

Вход: 1. h - число колец графа G_h .

2. n_1, n_2 - коды полуграниц графа G_h .

Выход: Дистанция графа $D(G_h)$ (число Винера).

Метод: 1. $D_{G_1}(u) \leftarrow 9$; $D_{G_1}(v) \leftarrow 9$; $\Phi \leftarrow 18$;

2. для $i + 1$ до $h-2$ шаг 1 цикл;

$C1 \leftarrow 4i + 4$; $C2 \leftarrow C1 + C1$; $C1 \leftarrow C1 + 4$;

если { i -й разряд n_1 равен 1} то

$$\left\{ \begin{array}{l} D_{G_{i+1}}(u) \leftarrow D_{G_i}(u) + C1; \\ D_{G_{i+1}}(v) \leftarrow D_{G_i}(u) + C2; \end{array} \right.$$

иначе

если { i -й разряд n_2 равен 1} то

$$\left\{ \begin{array}{l} D_{G_{i+1}}(u) \leftarrow D_{G_i}(v) + C2; \\ D_{G_{i+1}}(v) \leftarrow D_{G_i}(v) + C1; \end{array} \right.$$

иначе

$$\left\{ \begin{array}{l} D_{G_{i+1}}(u) \leftarrow D_{G_i}(u) + C2; \\ D_{G_{i+1}}(v) \leftarrow D_{G_i}(v) + C2; \end{array} \right.$$

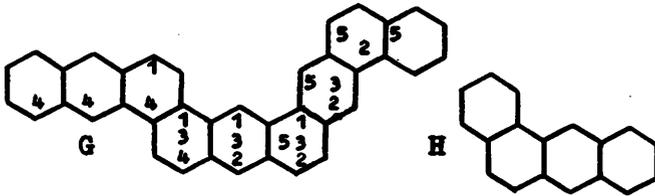
$$\Phi + \Phi + (D_{G_{i+1}}(u) + D_{G_{i+1}}(v))$$

все;

$$D(G_h) + 2\Phi + 2(h-1)(6h+11) + 27;$$

Конец;

Из построения алгоритма 4 следует линейная по числу колец трудоемкость вычисления числа Винера графов класса K_6 по кодам полуграниц.



$$\begin{array}{l} n_1(G) = \overbrace{(0010100)}^{n_1(H)} \\ n_2(G) = \overbrace{(0100001)}^{n_2(H)} \end{array} \quad \begin{array}{l} n_1(H) = (10) \\ n_2(H) = (00) \end{array}$$

a)

$$\begin{array}{l} k=1 \\ n_1(H) (10) (00) \begin{array}{|c|c|} \hline 10 \\ \hline 10 \\ \hline \end{array} (0) \\ n_2(H) (00) (01) \begin{array}{|c|c|} \hline 00 \\ \hline 00 \\ \hline \end{array} (1) \end{array} \quad \begin{array}{l} k=2 \\ n_2(H) (00) (0010100) \\ n_1(H) (10) (0100001) \end{array}$$

$$\begin{array}{l} k=3 \\ n_1^*(H) (01) (001) \begin{array}{|c|} \hline 01 \\ \hline \end{array} (00) \\ n_2^*(H) (00) (010) \begin{array}{|c|} \hline 00 \\ \hline \end{array} (01) \end{array} \quad \begin{array}{l} k=4 \\ n_2^*(H) (00) (\begin{array}{|c|} \hline 00 \\ \hline \end{array} 101 \begin{array}{|c|} \hline 00 \\ \hline \end{array}) \\ n_1^*(H) (01) (\begin{array}{|c|} \hline 01 \\ \hline \end{array} 000 \begin{array}{|c|} \hline 01 \\ \hline \end{array}) \end{array}$$

б)

Рис. 12

Вложение и пересечение графов. Пусть граф G состоит из h колец и граф H имеет h' колец, $h' \leq h$. Граф H вкладывается в граф G , если в G существует подграф $H' \subseteq G$ такой, что $H \cong H'$. Если граф H вкладывается в граф G , то полуграницы графов H и H' совпадают. Любое подслово кодов полуграниц $n_1(G)$ и $n_2(G)$ образует коды полуграниц графа, получаемого из G удалением колец, соответствующих отброшенным элементам n_1 и n_2 . В алгоритме определения вложения графа H в граф G проверяется совпадение кодов графа H со всеми подсловами длины h' в кодах полуграниц графа G . На рис.12а показан пример вхождения кода графа H как подслова в код графа G .

Алгоритм 5. Определение вложения графа

Вход: h - число колец графа G , коды $n_1(G)$, $n_2(G)$,

h' - число колец графа H , коды $n_1(H)$, $n_2(H)$,
 $n_1^*(H)$ и $n_2^*(H)$.

Выход: Номера колец графа G , образующих подграф, изоморфный H .

Метод: $COD(1) \leftarrow n_1(H)$; $COD(2) \leftarrow n_2(H)$;

$COD(3) \leftarrow n_2(H)$; $COD(4) \leftarrow n_1(H)$;

$COD(5) \leftarrow n_1^*(H)$; $COD(6) \leftarrow n_2^*(H)$;

$COD(7) \leftarrow n_2^*(H)$; $COD(8) \leftarrow n_1^*(H)$;

для $k \leftarrow 1$ **до** 4 **шаг 2 цикл;**

$m_1 \leftarrow COD(k)$;

$m_2 \leftarrow COD(k+1)$;

для $i \leftarrow 1$ **до** $h - h' + 1$ **шаг 1 цикл;**

если

$\left\{ \begin{array}{l} m_1 \text{ является подсловом в } n_1(G), \\ \text{начиная с } i\text{-го элемента } n_1(G), \end{array} \right\}$

и m_2 является подсловом в $n_2(G)$,
 (начиная с i -го элемента.

то
 { подграф, состоящий из колец $i, i+1, \dots, i+h'-1$,
 }
 { изоморфен графу H .
 }

все;

все;

Конец;

На рис. 12б показан пример работы алгоритма 5 для графов на рис. 12а. Для всех итераций внешнего цикла выделены вхождения кодов графа H как подслов кодов G , соответствующие пяти различным вложениям графа H . Кольца графа G помечены порядковым номером вложения. Из построения алгоритма 5 следует, что его трудоемкость составляет $O(k(h-h'+1))$, где k - трудоемкость проверки вхождения кода графа H как подслова в код

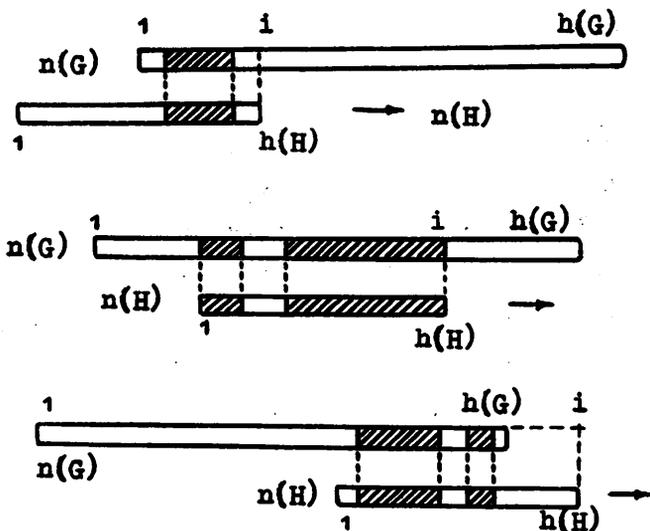


Рис. 13

графа G . В частности, из этой оценки следует трудоемкость определения изоморфизма графов $O(1)$ при $h = h'$.

Под пересечением двух графов G и H будем понимать максимальный по числу колец граф G' класса K_6 такой, что G' вкладывается в G и H . Если в алгоритме вложения графа проверяется вхождение кода графа H как подслова в код графа целиком, то в алгоритме пересечения определяются совпадающие участки кодов G и H и выбираются максимальные. На рис.13 показаны возможные способы сравнения кодов графов G и H , совпадающие участки кодов заштрихованы. В алгоритме 6 находятся все возможные общие части графов G и H , из которых выбираются содержащие максимальное число колец. Подслова в кодах G и H , выбираемые для анализа совпадающих частей, определяются в соответствии со схемой на рис.13.

Алгоритм 6. Определение пересечений графов

Вход: 1. Граф G с числом колец h , коды $n_1(G), n_2(G)$;
 2. Граф H с числом колец h' , $h' \leq h$, коды $n_1(H), n_2(H)$ и $n_1^*(H), n_2^*(H)$.

Выход: 1. МАКС - число колец в пересечениях.
 2. КОЛ - количество всех пересечений.
 3. ПЕРЕС - множество всех пересечений.

Элемент ПЕРЕС представляется в виде

$\{(i_1, i_{\text{макс}}), (j_1, j_{\text{макс}})\}$, где $i_1, i_2, \dots, i_{\text{макс}}$ -

кольца графа G , $j_1, j_2, \dots, j_{\text{макс}}$ - кольца графа H ,

соответствующие пересечению.

Метод: $COD(1) \leftarrow n_1(H)$; $COD(2) \leftarrow n_2(H)$;
 $COD(3) \leftarrow n_2(H)$; $COD(4) \leftarrow n_1(H)$;
 $COD(5) \leftarrow n_1^*(H)$; $COD(6) \leftarrow n_2^*(H)$;

$COD(7) \leftarrow n^*(H); \quad COD(8) \leftarrow n_1^*(H);$
 $МАКС \leftarrow 0; \quad КОЛ \leftarrow 0; \quad ПЕРЕС \leftarrow \emptyset;$

для $k \leftarrow 1$ до 4 шаг 2 цикл;

$m_1 \leftarrow COD(k);$

$m_2 \leftarrow COD(k+1);$

{ определение пересечений
кодов n_1, n_2 и m_1, m_2 и формирование
МАКС, КОЛ, ПЕРЕС. }

все;

$МАКС \leftarrow МАКС + 2$

Конец;

В представленной ниже части алгоритма выделяются подслова графов G и H и подсчитываются наибольшие совпадающие участки кодов. Пересечения максимального к текущему шагу алгоритма порядка ТЕК_МАКС накапливаются в множестве ТЕК_ПЕРЕС до появления пересечения большего порядка, l есть длина выделенных подслов, в которых сравниваются разряды.

$h \leftarrow h-2; \quad h' \leftarrow h'-2;$

для $i \leftarrow 1$ до $h + h' - 1$ шаг 1 цикл;

если $i < h'$ то

{ в n_1 и n_2 выделить подслова r_1 и r_2 , состоящие из разрядов $(1, 2, \dots, i)$, а в m_1 и m_2 - s_1 и s_2 из разрядов $(h'-i+1, \dots, h')$; $l \leftarrow i;$ }

если $h' \leq i \leq h$ то

{ в n_1 и n_2 выделить подслова r_1 и r_2 , состоящие из разрядов $(i-h'+1, \dots, i)$, в качестве s_1 и s_2 взять m_1 и m_2 ; $l \leftarrow h';$ }

если $i > h$ то

{ в n_1 и n_2 выделить r_1 и r_2 , состоящие из разрядов $(i-h'+1, \dots, h)$, а в m_1 и m_2 выделить s_1 и s_2 из разрядов $(1, \dots, h-h'-i)$, $l \leftarrow h+h'-i;$ }

$ТЕК_ПЕРЕС \leftarrow \emptyset;$

ТЕК_МАКС ← 0;

для j ← 1 до l шаг 1 цикл;

если { j-й элемент в r₁ и r₂ равен j-му элементу }
в s₁ и s₂ соответственно;

то { ТЕК_МАКС ← ТЕК_МАКС + 1;
ТЕК_ПЕРЕС ← ТЕК_ПЕРЕС ∪ {j}; }

иначе

если ТЕК_МАКС=МАКС то { восстановить в ТЕК_ПЕРЕС нумерацию колец для G и H;
ПЕРЕС ← ПЕРЕС ∪ ТЕК_ПЕРЕС;
КОЛ ← КОЛ + 1; }

иначе

если ТЕК_МАКС > МАКС то { восстановить в ТЕК_ПЕРЕС нумерацию колец для G и H;
ПЕРЕС ← ТЕК_ПЕРЕС;
МАКС ← ТЕК_МАКС;
КОЛ ← 1; }

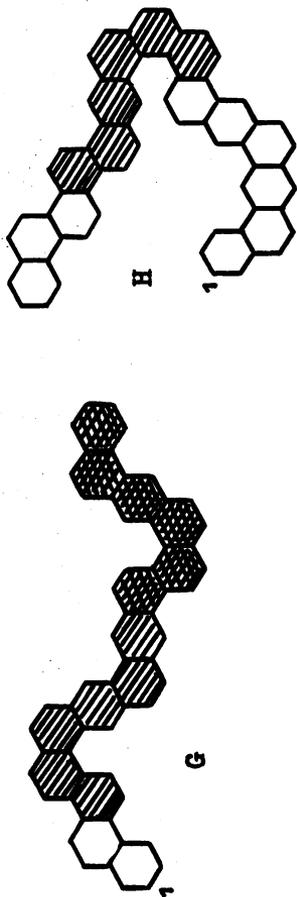
ТЕК_МАКС ← 0;

ТЕК_ПЕРЕС ← ∅;

все j;

все i;

На рис.14 приводится пример работы алгоритма 6 для половин сравнения кодов, при которых находятся пересечения графов. При определении пересечения G и H выделено два соответствия: кольца 7,8,...,12 графа H соответствуют кольцам 3,4,...,8 и 9,10,...,14 графа G. Трудоемкость алгоритма 6 составляет $O(kh'(h+h'))$, где k - трудоемкость выделения под слова заданного размера.



$$n_1 = (010001001100)$$

$$n_2 = (101100010001)$$

$k = 1$

$$n_1 \begin{pmatrix} 0100 & 01 & 00 & 1100 \\ (1 & 01 & 01 & 1100 & 101) \end{pmatrix}$$

$$m_1 \begin{pmatrix} 1011 & 00 & 01 & 0001 \\ (0 & 00 & 10 & 0001 & 010) \end{pmatrix}$$

$$m_1 = (101011100101)$$

$$m_2 = (000100001010)$$

$k = 2$

$$n_1 \begin{pmatrix} 01 & 0001 & 001 & 100 \\ (000 & 10 & 0001 & 010) \end{pmatrix}$$

$$m_2 \begin{pmatrix} 10 & 1100 & 010 & 001 \\ (101 & 01 & 1100 & 101) \end{pmatrix}$$

Рис. 14. Иллюстрация к алгоритму нахождения общих частей графов

Заключение

В работе описан алгоритм конструктивного перечисления графов неразветвленных ката-конденсированных гексагональных систем. Алгоритм основан на новом представлении графа этого класса с помощью пары графов специального вида, которые, в свою очередь, представляются двоичными наборами. Алгоритм генерации графов имеет следующие особенности: проверка соответствия пары произвольных двоичных наборов некоторому графу данного класса и определение изоморфности очередного сгенерированного графа уже полученным графам осуществляются за константное число операций; легкость прекращения и возобновления процесса генерации; компактность представления графов; простота алгоритма и программы генерации; порождаемые графы не обязательно вложимы в правильную гексагональную решетку на плоскости. Рассмотрены основанные на введенном представлении графов алгоритмы определения вложимости графа в гексагональную решетку, восстановления матрицы смежности, визуализации графов, определения вложения и пересечения (максимальной общей части) графов, вычисления дистанции графа (индекса Винера). Таким образом, создана алгоритмическая основа для эффективной генерации классов графов неразветвленных ката-конденсированных гексагональных систем и исследования их свойств.

Л и т е р а т у р а

1. ФАРАДЖЕВ И.А. Конструктивное перечисление комбинаторных объектов //Алгоритмические исследования в комбинаторике. - М.: Наука, 1978. - С. 3-11.
2. BALABAN A.T., BRUNVOLL J., CIOSLOWSKI J. a.o. Enumeration of Benzenoid and Coronoid Hydrocarbons //Z.Naturforsch. - 1987. -Vol. 42a. - P. 863-870.
3. МЖЕЛЬСКАЯ Е.В., СКОРОБОГАТОВ В.А. Применение теории графов в химии полициклических бензоидных углеводородов. - Новосибирск, 1987. - 34 с. (Препринт/АН СССР, Сиб. отд-е Ин-т математики, № 35).

4. МЖЕЛЬСКАЯ Е.В. Графы полициклических соединений //Вопросы алгоритмического анализа структурной информации. -Новосибирск, 1987. - Вып. 119: Вычислительные системы. -С. 71-90.
5. BALABAN A.T., HARARY F. Enumeration and proposed nomenclature of benzenoid cata-condensed polycyclic aromatic hydrocarbons //Tetrahedron. - 1968. -Vol. 24. - P. 2505-2516.
6. KNOP J.V., MULLER W.R., SZYMANSKI K., TRINAJSTIĆ N. Computer generation of certain classes of molecules. - Zagreb: SKTN/Kemija u industriji, 1985. - 166 p.
7. BALABAN A.T. Configurations of annulenes //Tetrahedron. - 1971. -Vol. 27, N 23-24. -P. 6115-6131.
8. JOVANOVIĆ A.D. A basis for characterization of hexagonal animals //Proc. Fourth Yugoslav Seminar on Graph Theory, Novi Sad: 1983. - P. 167-175.
9. HE W., HE W. Generation and enumeration of planar polycyclic aromatic hydrocarbons //Tetrahedron. - 1986.-Vol. 42, N 19. - P.5291-5299.
10. TRINAJSTIĆ N., JERICEVIĆ Z., KNOP J.V., MULLER W.R., SZYMANSKI K. Computer generation of isomeric structure //Pure and Appl. Chem. - 1983. - Vol. 55. - P. 379-390.
11. КОРНИЛОВ М.Ю., ЗАМКОВОЙ В.И. Расчет числа изомеров и кодирование конденсированных полициклических систем: перечисление полибензолов //Теорет. и эксперим. химия. - 1987. - № 1. - С. 54-61.
12. STOJMEŃOVIĆ I., TOŠIĆ R. Generating and counting hexagonal systems //Graph Theory: Proc./Sixth Yugoslav Seminar on Graph Theory, Dubrovnik, april 1985. - Novi Sad: 1986. -P.189-198.
13. JOVANOVIĆ A.D. Combinatorial characterization of hexagonal systems //Discrete Appl. Math. - 1988. -Vol.19. -P.259 - 270.
14. TOŠIĆ R., KOVAČEVIĆ M. Generation and Counting Unbranched Catacondensed Benzenoids //J.Chem. Inf. Comput. Sci.-1988. - Vol. 28. -P. 29-31.
15. CYVIN B.N., BRUNVOLL J., CYVIN S.J. All-Benzenoid Systems: Enumeration and Classification of Benzenoid Hydrocarbons. VI //Math. chem. (MATCH). - 1988. - N 23. - P. 163-173.
16. HE W., HE W., CYVIN B.N., CYVIN S.J.,BRUNVOLL J. There are Exactly Ninety-Eight Concealed Non-Kekuléan Benzenoids with Twelve Hexagons. Appendix Benzenoids with Hexagonal Symmetry and Fourty-Nine Hexagons //Ibid. - P.201-207.

17. BALABAN A.T., BRUNVOLL J., CYVIN B.N., CYVIN S.J. Enumeration of Branched catacondensed benzenoid hydrocarbons and Their numbers of Kekulé Structures //Tetrahedron. -1988.-Vol. 44, N 1. - P. 221-228.
18. ХАРАРИ Ф. Теория графов. - М.: Мир, 1973. - 300 с.
19. GUTMAN I., MARKOVIĆ S., LUKOVIĆ U., RADIVOJEVIĆ V., RANČIĆ S. On Wiener numbers of benzenoid hydrocarbons //Zbor - nic Radova Prirodno-matematičkog fakulteta u Kragujevcu. -1987. - Vol. 8. - P. 15-34.
20. SKOROBOGATOV V.A., DOBRYNIN A.A. Metric Analysis of Graphs //Math. Chem. (MATCH). - 1988. - N 23. -P. 105-151.
21. РУВРЭ Д.Г. Следует ли заниматься разработкой топологических индексов? //Химические приложения топологии и теории графов. - М.: 1987. - С. 183-205.
22. Его же. Химию прогнозирует топология //В мире науки. - 1986. № 11. - С. 14-22.
23. БОНЧЕВ Д.Г. Характеризация химических структур с помощью теории информации и теории графов: Автореф. дис. докт. хим. наук: 02.00.03, 02.00.04. - Бургас, 1983. - 48 с.
24. ДОБРЫНИН А.А. Дистанция молекулярных графов полициклических соединений //Анализ данных в экспертных системах.-Новосибирск, 1986. -Вып. 117: Вычислительные системы. -С. 114-122.

Поступила в ред.-изд.отд.

15 марта 1989 года