

СИСТЕМА АВТОМАТИЗАЦИИ РЕШЕНИЯ ЗАДАЧ НА ГРАФАХ

Г.Н.Белоглазов, В.А.Скоробогатов

В ряде прикладных областей обработки информации возникает необходимость решения задач на графах, поскольку графы используются как модели различных дискретных систем. Интересным в этой связи является создание специальной системы математического обеспечения, предназначенной для решения задач на графах, которая могла бы взаимодействовать с общей системой математического обеспечения ЭВМ.

В работе описан вариант системы ГРАФ, реализованный на ЭВМ "Минск-32", при разработке которой использован модульный принцип. Модули написаны на языке ФОРТРАН [1] и автокоде ЯСК [2], при этом использовались некоторые стандартные программы, входящие в состав общей СМО "Минск-32".

1. Система ГРАФ состоит из следующей совокупности блоков: генерации графов, ввода, преобразования входного представления, коррекции входной информации, разбиения множества вершин на классы эквивалентности, операции над графами, блока графики, служебных программ. Блоки, в свою очередь, содержат набор модулей.

Структуру системы ГРАФ можно представить в виде ориентированного графа, у которого вершины помечены номером блока (рис.1).

Все модули связаны глобальной общей областью (оператор ФОРТРАНа COMMON), в которой находится информация о числе вершин, числе ребер, степенях вершин и матрицах смежности двух графов. Это позволяет по мере необходимости добавлять в систему новые модули, строить блоки и программы для решения требуемой задачи. При помощи системы ГРАФ можно обрабатывать графы ориентированные и неориентированные, с петлями, мультиграфы, с раскрашенными вершина-

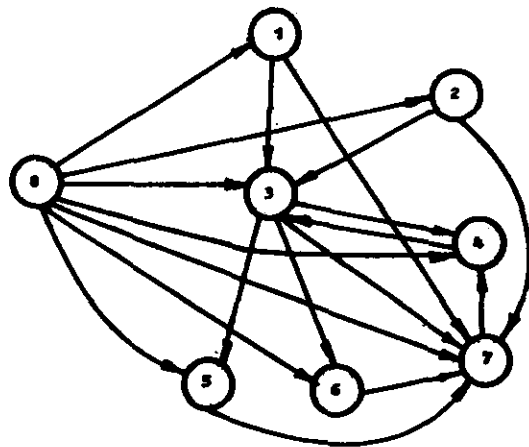


Рис.1

ми. При этом максимальное число вершин может достигать пяти-сот, максимальная степень вершини - трех-сот, максимальное число ребер - семи с половиной тысяч.

Управление системой на ЭВМ "Минск-32" осуществляется при помощи диспетчера [3], при этом работой № 1 является последовательное выполнение некоторой заданной совокупности модулей системы, а работой № 2 является выполнение

по мере необходимости сервисной программы, которая состоит из служебных программ, входящих в СМО "Минск-32".

2. Описание графов.

2.1. ОПРЕДЕЛЕНИЯ. Пусть задано конечное множество  $X = \{x_1, x_2, \dots, x_n\}$  из  $n$  элементов  $x$ ,  $X = \{x_i / i = \overline{1, n}\}$ , где  $x_i$  есть  $i$ -я вершина графа. На множестве  $X$  введем отношение смежности. Пусть множество двуместных предикатов  $P = \{P_{ij} / ij = \overline{1, n}\}$  таких, что  $P_{ij}(x_i, x_j) = 1$ , если  $x_i$  смежно с  $x_j$  и  $P_{ij}(x_i, x_j) = 0$ , если  $x_i$  не смежно с  $x_j$ . Отметим, что для неорграфов  $P_{ij} = P_{ji}$ . Пара смежных вершин называется ребром.

В каждом конкретном случае множества  $X$  и  $P$  должны быть заданы каким-то способом, например, простым перечислением. Форма перечисления может быть разной: списки, матрицы и т.д.

2.1.1. Пример. Пусть  $X = (x_1, x_2, x_3, x_4)$ ,  $P = \{P_{12}, P_{14}, P_{23}, P_{34}\}$ . Указанные здесь предикаты равны единице. Отсюда следует, что пары вершин  $(x_1, x_2)$ ,  $(x_1, x_4)$ ,  $(x_2, x_3)$ ,  $(x_3, x_4)$  смежны (см. рис.2).

Очевидно, что для представления графа в ЭМ этой информации достаточно, но поскольку такая запись для человека не представляется удобной, то обычно граф изображается в виде диаграммы.

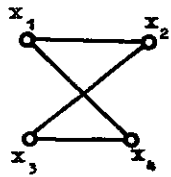


Рис.2

Заметим, что изображение графа называют его геометрической реализацией. Очевидно, что один и тот же граф может иметь много геометрических реализаций.

Под цепью  $(x_1, x_j)$  будем понимать последовательность ребер, начинающуюся в вершине  $x_1$  и заканчивающуюся в вершине  $x_j$ , в которой каждый конец ребра служит началом следующего ребра. В цепи каждое ребро может встречаться только один раз.

Говорят, что

а) При вершине  $x$  графа  $G$  есть петля, если  $P_{xx} = 1$  и  $P_{xx} \in P$ . Петель при вершине может быть несколько, в этом случае  $P_{xx} = n$ , где  $n$  - число петель. Графическая интерпретация данного положения показана на рис. 3 и 4.

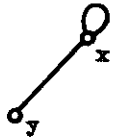


Рис.3



Рис.4



Рис.5



Рис.6

б) Пара вершин  $x, y$  связана мультиребром кратности  $\mu$ , если  $P_{xy} = \mu$ ; максимальное число  $\mu$  для графа  $G$  также называют мультичислом графа  $G$  или цветом ребра (рис. 5).

в) Вершина имеет цвет, если хотят ввести некоторый дополнительный признак, вес и т.д., чтобы отметить ее особые свойства. Цвет определяется натуральным числом.

г) Ориентированный граф отличается от неографа тем, что имеет место  $P_{xy} \neq P_{yx}$ , т.е. если  $x$  соединена дугой с  $y$ , то отсюда не следует, что  $y$  соединена с  $x$  (рис.6). Исключение составляет петля, так как  $P_{xx} = P_{xx}$  для ориентированных графов всегда выполняется.

2.2. Графы, представимые при помощи вышеуказанных понятий, могут быть описаны символами, приведенными в табл. I, а также символами X - крест, K - конец описания. Символ последних будет ясен из дальнейшего.

Описываемый объект	Свойства объекта	Обозначения	Количественные характеристики
Вершина	Номер	Н	I - 500
Вершина	Цвет	Ц	0 - 99
Ребро	Число ребер	P	0 - 99
Дуга	Число дуг	Д	0 - 99
Петля	Число петель	Ц	0 - 99

Графы в системе представляются в виде совокупности покрывающих их цепей. Возможность этого основана на очевидном утверждении о том, что граф представим своим покрытием некоторой совокупностью цепей. Такая совокупность содержит символы всех вершин и ребер, причем каждое ребро может содержаться только один раз, в то время как вершины могут повторяться неограниченное число раз.

2.2.2. Примеры описания графов.

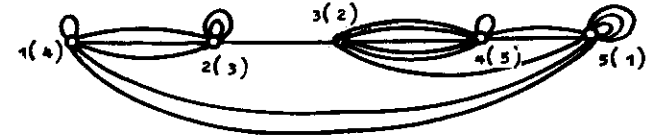


Рис. 7

Языковое описание графа на рис. 7 имеет вид:

НЦ4ПР3Н2Ц3П2Р1Н3Ц2ПОР5Н4Ц5ПР2Н5Ц1П3Р1Н3,Н5Р2Н1,К.

Числа в скобках обозначают цвет вершины. Цвета и петли можно повторно не описывать после того, как были описаны ("пройдены") все вершины графа. Если граф ориентированный, то вместо символа P следует пользоваться символом Д. В случае, если граф без петель и несокращенный, то символами Ц и П можно не пользоваться.

Если на место некоторого символа был случайно записан лишний символ или символ, не принадлежащий языку, для коррекции в процессе кодирования используется символ X. В случае пропуска символов поступают аналогично: H25HHC1HHCOR5HHC1H10...

Описанное выше представление графа будем в дальнейшем называть символьным представлением.

2.3. Для упрощенного описания обыкновенных неокрашенных неографов может быть использован способ, основанный на покрытии графа цепями. Например,

- числа: -7 - раздатель, цепи (соответствует зашпоз);
- 8 - конец цепи массива цепей;
- 88 - конец массива цепей (соответствует концу описания графа).



Рис.8

Ребро, как и ранее, может быть пройдено только один раз, вершина может быть пройдена многократно.

Следующая запись иллюстрирует упрощенный способ описания (рис.8):

-07001002003005-07002004003006-07004001003-07-77.

что соответствует цепям 1, 2, 3,5; 2,4,3,6; 4,1,3.

Указанный способ описания графов на практике позволяет избежать большого числа ошибок и интуитивно кажется более естественным для применения его человеком.

2.4. После ввода в машину закодированных цепей, граф представляется в виде матрицы смежности, либо в виде списка (или кодовой реализации). Каждый элемент списка занимает одну ячейку оперативной памяти. Для больших графов список размещается на магнитной ленте в нескольких зонах стандартной длины.

Матрица смежности представляет собой квадратную матрицу, состоящую из нулей и единиц. Элементу матрицы соответствует разряд ячейки ЭМ.

В случае использования символьного представления одним из указанных машинных представлений сопровождается таблица, в которой находится информация о раскраске, наклоне петель и мультиребер.

3. Состав системы ГРАФ. Перечень модулей, их идентификаторы и принадлежность блокам системы представлены в табл. 2. Ниже приводится описание назначения модулей и их функциональных особенностей по блокам.

Таблица 2

Перечень идентификаторов программ и состав блоков

Генерация 1	Ввод 2	Преобразование 3	Коррекция 4	Разделение 5	Операции 6	Графика 7
ГОЛУБЬ; <РАВ> <САНДЛ> <ПЕС...>	ВСТ; <СИМ> ВЧП; ФОРМСТ	ФОРТА; <ФОРМАЦ> ПОДР; ПОР. ПОРТ ФМАТ	КОРР; <КМ> КОРВК; КОС1; КОС2	ЛАВРЕ АРУС <РАВ> <САМ> Л.АМПУЛ	БОПЕР; <ДСУМА> ДЕТРО КОМП1, КОМП2 ДОПОЛ КОННО ДИЗМО ШТРИХ ГАММА СОБД КОРН1, КОРН2, РТРАФ	ВАНГО; <ЛАВРЕ> ВВВОД; GRAPE; <ДЕФОР> ФОРМА; НОМЕТ; ТЕБЕСТ; LINE; ОР
ПЕРМС, UNIT ZERO ЗАММ ВОММ ВБОМ ЛМ, ДИЛ	ЗАММ ДИЛ ЛМ	УТОР ПЕРМС ДИЛ; ЗАММ ЛМ, UNIT, ZERO	ДИЛ; ЗАММ ЛМ	ДИЛ, ЗАММ УТОР, ПОР СМСМ, ПЕСТЬ, ПЕСТИ, СЕТАБ, СТЕОМ ПОЖ, ПОЖК, СРМ, ПССМС	ДИЛ, ЗАММ ЛМ, UNIT 1 UNIT2, ZERO1 ZERO 2 ПЕРМС, ПОМР1, ПОМР2 ПОМР3	ДИЛ; ЗАММ ЛМ
Следующий блок						

\* Подчеркнутые двумя чертами (СИМ) идентификатор означает, что соответствующая программа реализована на ЯОКЕ.

3.1. Блок генерации служит для порождения графов в системе.

3.1.1. Модуль ГСЛУГ (генератор случайных графов) может использоваться для экспериментальной проверки алгоритмов, он позволяет получать серии случайных графов, принадлежащих некоторому классу, параметры графов (число вершин, ребер, число графов в серии) могут быть заданы извне.

3.1.2. Модуль СИМПГ (синтез графа в символьном представлении) совместно с блоком коррекции служит для обеспечения возможности ввода в ЭВМ сложных графов.

Вначале задаются число вершин  $P$  и число ребер  $Q$ , затем строится звездный граф  $K_{1, P-1}$  [4] с заданным числом вершин и добавляется требуемое количество ребер. Граф представляется в виде цепей в символьном представлении при помощи подпрограммы перевода в символы, затем выводится на графопостроитель. При помощи полученной диаграммы можно сконструировать желаемый граф с помощью блока коррекции, т.е. можно удалить ребра, вершины, раскрасить вершины, добавить петли, заменить мультиребра и т.д.

Для получения графа в числовом представлении можно использовать ГСЛУГ, применяя эту же методику.

3.2. Блок ввода служит для введения с перфокарт графов, описанных на входном языке.

3.2.1. Модуль СИМ приводит массив цепей, описанных на входном языке, к виду, удобному для коррекции. Если, например, фрагмент цепи представлен как НЦЦ5ПОР3Н2..., то после работы модуля он приобретает вид: НОЦЦО5ПООРОЗН002.... Если при описании были допущены синтаксические ошибки или использованы символы, не принадлежащие входному языку, то на устройство печати выдается сообщение с указанием координат ошибок. Результирующий массив записывается на МЛ.

3.2.2. Модуль ВГЧП предназначен для ввода с перфокарт массива цепей, заданных в числовом представлении, и включает в себя подпрограмму ФОМСТ, которая формирует массив использованных номеров и массив степеней и вычисляет количество вершин и ребер вводимого графа. Массив цепей записывается на магнитную ленту с именем, которое можно изменять по усмотрению пользователя с помощью служебной программы ЗАМИМ.

3.3. Блок преобразования входного представления состоит из трех основных модулей: КОДР, ФМАТ, ФОРТА.

3.3.1. Модуль КОДР формирует из массива цепей список (кодированную реализацию). Алгоритм работы программы КОДР двухпросмотровый. За первый просмотр формируются массив степеней, массив использованных номеров и массив адресов списка с указанием номера зоны на магнитной ленте, в которой помещена строка списка соответствующей вершины. За второй просмотр формируется список и помещается в зону, указанную в массиве адресов.

3.3.2. Модуль ФМАТ формирует из массива цепей матрицу смежности. Программа имеет два режима: обычный и режим операции. В обычном режиме формируются матрицы до максимальной размерности, т.е. до 500. В режиме операций формируются матрицы смежности графов, являющихся операндами для некоторых булевых операций, в этом случае размерность матриц - до 250. В процессе формирования матриц программа выдает сообщения на печать о повторяющихся ребрах в цепях. По желанию можно получить распечатку матрицы смежности при помощи служебного модуля ПЕЧМС.

3.3.3. Модуль ФОРТА (формирование таблицы) использует как подпрограммы следующие модули: ФОРМЦ, КОДР, ПОР, ПОРТ.

3.3.3.1. Модуль ФОРМЦ производит формирование массива использованных номеров, массива цветов, массива петель, массива мультиребер и массива цепей в числовом представлении. Программа работает с откорректированным массивом цепей в символьном представлении. За один просмотр происходит расшифровка символов и перевод в числовое представление. Массивы мультиребер и цепей записываются на магнитную ленту.

3.3.3.2. Модуль ПОР упорядочивает по убыванию те части строк таблицы, в которых находятся мультичисла.

3.3.3.3. Модуль ПОРТ производит полное лексикографическое упорядочение сформированной таблицы. Модуль использует служебную подпрограмму УПОР.

3.4. Блок коррекции состоит из двух модулей: КОРР, КОРРМ.

3.4.1. Модуль КОРР предназначен для коррекции массива цепей в символьном представлении и включает в себя подпрограммы КЛМ, КОРВК, КОРЛГ, КОРС2.

3.4.1.1. Модули КЛМ и КОРВК служат для ввода корректировочной информации. В случае малого объема корректировочной информации используется модуль КЛМ ввода корректировочной информации с пультавой машинки в режиме "подсказок". При большом объеме корректировочной информации используется модуль КОРВК, который вводит

массив корректировочной информации с перфокарт, расшифровывает и распределяет его по соответствующим массивам.

3.4.1.2. Модуль KOPC1 производит удаление вершин, замену цвета вершины, замену числа петель и изменение числа ребер (дуг).

3.4.1.3. Модуль KOPC2 производит добавление вершин и ребер (дуг).

3.4.2. Модуль KOPPM служит для коррекции матрицы смежности и содержит следующие модули как подпрограммы: УДАВ - удаление вершин; УДАР - удаление ребра; ДОБРЕ - добавление ребра.

Ввод корректировочной информации производится в режиме диалога с пультовой машинки, при этом выполняется одна из трех вышеназванных подпрограмм, работа которых состоит в том, что в требуемое место матрицы смежности заносится 0 или 1.

3.5. Блок разбиений множества вершин графа на классы эквивалентности состоит из следующих модулей:

3.5.1. Модули LABRE и ЯРУСН позволяют получать  $\lambda$ -разбиение [5] графа в зависимости от машинного представления (матрица смежности или список).

3.5.2. Модуль LPA3 производит разбиение по  $\lambda$ -матрице [5]. Использует модуль получения  $\lambda$ -матрицы LAM и модуль ее лексикографического упорядочения LAMUP.

3.5.3. Модуль ГОТН реализует разбиение из [6].

3.5.4. Модуль ТРИПД осуществляет измельчение  $\lambda$ -разбиения. Использует модуль LABRE.

3.5.5. Модуль PA3T производит разбиение по упорядоченной таблице.

3.5.6. Модуль PA3PT позволяет получить разбиение множества вершин регулярного графа. Использует модуль ПССМС.

3.6. Блок операций служит для конструирования сложных графов из простых путем применения к ним булевых операций и содержит две-

надцать модулей, реализующих операции, приведенные в [7] (см. табл. 3).

Работа с блоками происходит в диалоговом режиме, т.е. с пультовой машинки можно задать желаемую операцию. Результирующий граф, представленный матрицей смежности, записывается на магнитную ленту. К блоку операций можно обращаться неограниченное число раз.

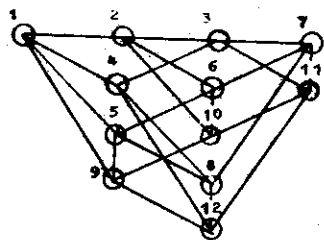


Рис. 9

Идентификатор	Операция	Обозначение
ДСУМ	декартова сумма	$G \times H$
ДЕПРО	декартово произведение	$G \times H$
КОМ1 1	композиция (или лексикографическое произведение)	$G[H]$
КОМ1 2		$H[G]$
ДОПОЛ	дополнение	$\bar{G}$
КОМБ	комбинция	$G \wedge H$
ДИЗЮ	дизъюнкция	$G \vee H$
ИТРИХ	итрих Шейфера	$G/H$
ГАММА	$\gamma$ -произведение	$G \cdot H$
СОЕД	соединение [11]	$G + H$
КОРН 1	корона	$G \circ H$
КОРН 2		$H \circ G$
РГРАФ	реберный граф	$L(G)$

На рис.9 приведен пример выполнения операций  $G \times H$ , где, согласно [4],  $G = K_3$  - треугольник,  $H = C_4$  - цикл из четырех вершин.

3.7. Блок графики позволяет получать изображение графов в виде слоев [8] по отношению к некоторой вершине на графопостроителе "Вектор-1301" [9] и состоит из восьми модулей.

3.7.1. Модуль FANCO предназначен для формирования массива номеров и координат вершин графа, представленного в виде  $\lambda$ -разложения. По  $\lambda$ -разложению графа модуль определяет номера вершин и связи ребер между вершинами соседних подмножеств, ребра внутри каждого из подмножеств и формирует массив номеров и координат соответствующего  $\lambda$ -разложения.

3.7.2. Модуль DEFOR по количеству подмножеств в  $\lambda$ -разбиении графа и по числу вершин в максимальном подмножестве (т.е. наименьшее наибольшее число вершин по отношению к остальным подмножествам) вычисляет константу формата или модуль ФОРМА. Расстояния между вершинами, между подмножествами вершин, между ребрами, соединяющими вершины внутри подмножества, выбраны таким образом, чтобы можно было нарисовать  $\lambda$ -разложение графа с 500-ми вершинами на листе бумаги формата А0.

3.7.3. Модуль ФОРМА обеспечивает рисование форматных линий по заданной константе формата (все форматы соответствуют ВСКД).

3.7.4. Модуль НОМЕР производит раскодирование заданного номера вершин и обеспечивает его рисование. Цифры закодированы так же, как графические символы программы.

3.7.5. Модуль ТЕКСТ обеспечивает рисование графических символов, обозначающих подпись или математическую формулу.

3.7.6. Модуль ЛИНЕ позволяет вычерчивать на графопостроителе прямые линии, произвольно расположенные на плоскости. Прямая линия задается координатами  $x$  и  $y$ , которые должны быть выражены числами элементарных перемещений пера графопостроителя и числом  $P$ , где значение  $P$  определяется следующим образом:

$$P = \begin{cases} 1 \rightarrow \text{первое перо опустить;} \\ 2 \rightarrow \text{второе перо опустить;} \\ 3 \rightarrow \text{перо поднять (первое или второе).} \end{cases}$$

3.7.7. Модуль ОКР обеспечивает вычерчивание на графопостроителе окружностей заданного рисунка. Величина радиуса задается целым числом элементарных перемещений пера.

3.8. Блок служебных программ.

3.8.1. Модуль УПОРС предназначен для упорядочения одномерных массивов по убыванию произвольных целых чисел. В основу работы подпрограммы положен алгоритм, описанный в работе [10].

3.8.2. Модуль УПОР применяется для лексикографического упорядочения таблиц и матриц, в работе которого используется модуль УПОРС. Различие заключается в том, что исходными массивами являются столбцы матриц или таблиц и при упорядочении очередного столбца предыдущая результирующая перестановка становится начальной. Процесс повторяется для всех столбцов. Переставляя строки таблицы или матрицы, согласно последней результирующей перестановке, получаем лексикографически упорядоченную исходную матрицу или таблицу.

3.8.3. Модули ДИАЛ, ВПМЦ, ВВОДП обеспечивают режим диалога.

3.8.4. Модули ПОИР1, ПОИР2, ПССМС, ПЕСТЬ, ПЕСТР, UNIT, UNIT1, UNIT2, ZKPO, ZKPO1, ZKPO2, ПОИР позволяют выполнять элементарные операции с матрицей смежности (перестановка строк, столбцов, занесение нуля или единицы и т.д.).

3.8.5. Модули СРМ, СРТАБ, СМСМ осуществляют сравнение матриц, таблиц и разбиений.

3.8.6. Модули ЛПМ, ЗАММ обеспечивают наиболее эффективный режим работы с магнитной лентой.

3.8.7. Модуль ПЕЧМС позволяет выводить на АЦПУ матрицы смежности.

#### 4. Возможные применения.

4.1. Широкое использование графов в качестве математических моделей систем вызывает необходимость разработки большого числа алгоритмов анализа и преобразований графов. При этом на практике по разным причинам не всегда удается достаточно хорошо оценить сложность разрабатываемых алгоритмов.

Блоки генерации и операций над графами позволяют автоматизировать процесс исследования алгоритмов, так как графы, полученные в системе, и результат некоторых операций над ними позволяют сделать определенные заключения об алгоритме.

4.2: Блок разбиений на классы эквивалентности позволяет решать достаточно широкий круг задач, где могут быть использованы указанные разбиения. К таким задачам, в частности, относится задача распознавания изоморфизма, для которой ниже приведена блок-схема конкретного состава системы, задачи таксономии в распознавании образов, задачи логического контроля схем в микроэлектронике и др. Сама по себе задача распознавания изоморфизма вызывает интерес у прикладников в связи с желанием иметь достаточно эффективный алгоритм. Примером реализации такого алгоритма является следующая программа.

4.2.1. Программа ISOM предназначена для распознавания изоморфизма графов с характеристиками, которые описаны в первом разделе. Результатом работы программы является либо заключение о неизоморфизме данных графов, либо подстановка, указывающая на изоморфизм.

Блок-схема программы приведена на рис. 10.

Работа программы происходит следующим образом. После трансляции ISOM помещается на ленту системы, на которой имеется все собранное и готовое к работе модули системы. Затем программа ISOM загружается в оперативную память и последовательно вызывает для выполнения своих функций модули согласно блок-схеме. Операторы имеют следующий смысл:

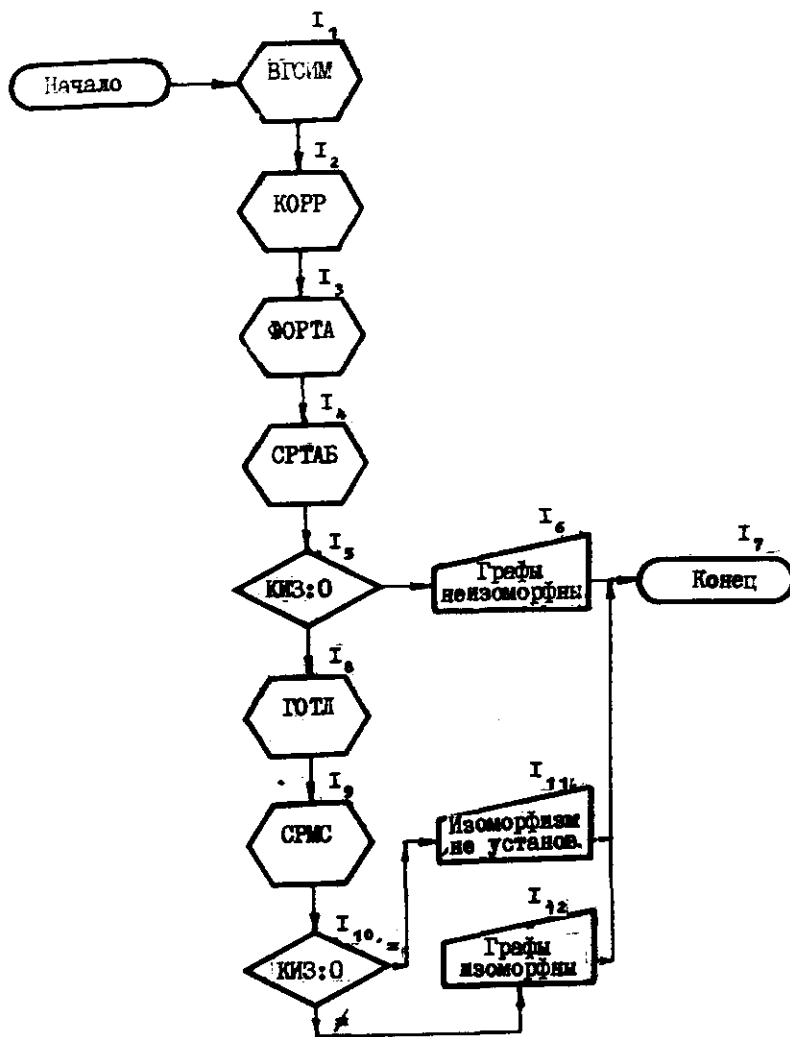


Рис. 10. Программа ISOM.

- I1 - происходит ввод цепей на входном языке;
- I2 - производится коррекция массива цепей, если обнаружены ошибки в кодировке;
- I3 - происходит формирование таблицы графов;
- I4 - сравниваются таблицы, и если произошло несовпадение, то на пультовую машинку выдается сообщение о том, что графы неизоморфны (I6), в случае совпадения с помощью модуля I8 множества вершин графов разбиваются на классы эквивалентности по совокупности признаков;
- I9 - происходит сравнение матриц смежностей с соответственно переставленными строками и столбцами, и результат выдается на пультовую машинку (I11, I12);
- I7 - конец.

4.2.2. Пример применения программы ISOM при контроле топологии интегральных схем. Часть задачи контроля топологии интегральных схем состоит в том, что необходимо найти структурные несоответствия полученной топологии с исходной принципиальной схемой. Для этого по топологии может быть восстановлена принципиальная схема, которая может содержать ошибки. Далее, представляя правильную и "ошибочную" схемы в виде графов, в процессе распознавания их изоморфизма можно установить появившиеся при проектировании ошибки.

Покажем, как эту задачу можно решать при помощи программы ISOM. Для этого рассмотрим способ представления принципиальной схемы в виде графа на примере принципиальной схемы, приведенной на рис.12.

Поставим в соответствие схеме вершиннораскрашенный неориентированный мультиграф следующим образом:

1. Транзистору соответствует вершина цвета 1 или 2 в зависимости от типа проводимости.
2. Шине земли соответствует вершина цвета 0 (нуль).
3. Емкости соответствует вершина цвета 3.
4. Резистору соответствует вершина цвета 4.
5. Диоду соответствует вершина цвета 5 или 6 в зависимости от типа.
6. Внешним узлом с постоянным источником питания соответствует вершины с цветами от 11 до 20.
7. Внешним узлом с переменными источниками соответствуют вершины с цветами от 21 до 50.

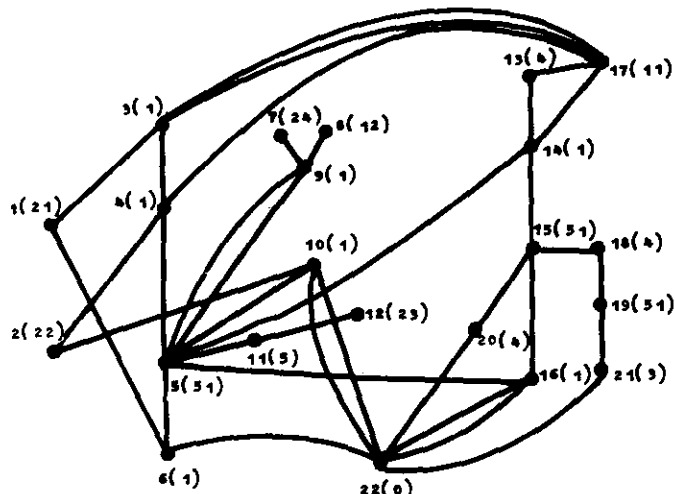


Рис. II

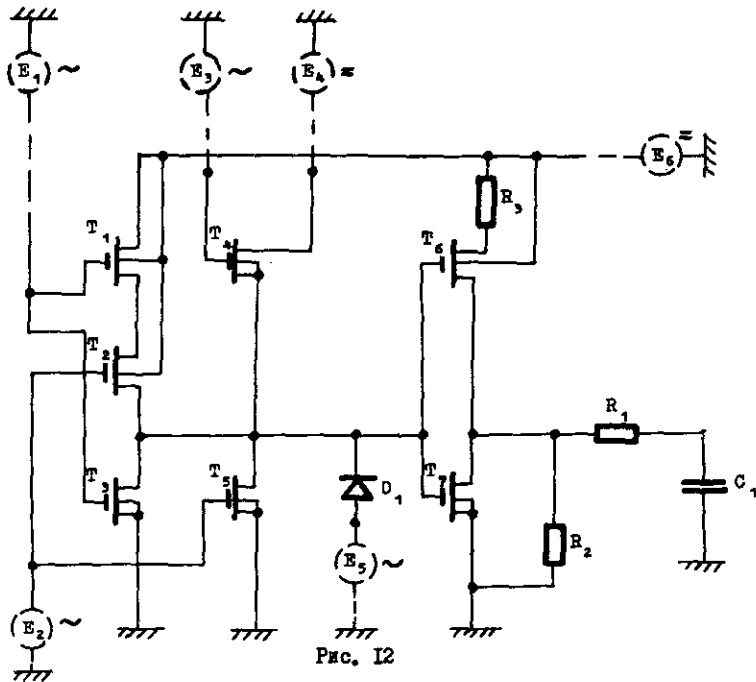


Рис. I2

8. Внутренним узлам соответствуют вершины цвета 5I.

9. Вершины соединим ребрами в соответствии с принципиальной схемой.

В результате получим граф (рис.II), соответствующий схеме, приведенной на рис.I2. Аналогичный граф можно получить при восстановлении принципиальной схемы по топологии. Далее задача поиска ошибки сведется к установлению неизоморфизма и специальным вопросам интерпретации найденных несоответствий.

#### Л и т е р а т у р а

1. СМО "МИНСК-32" Фортран, Е14, 069. 009II, Минск, 1973.
2. КУШНЕРОВ Н.Т., НЕМЕНМАН М.Е., ЦАГЕЛЬСКИЙ В.И. Программирование для ЭВМ "Минск-32". М., "Статистика", 1973.
3. БЕЛОКУРСКАЯ И.Л., КУШНЕРОВ Н.Т., НЕМЕНМАН М.Е. Диспетчер ЭВМ "Минск-32". М., "Статистика", 1973.
4. ХАРАРИ Ф. Теория графов. М., "Мир", 1973.
5. СКОРОВОГАТОВ В.А. О распознавании изоморфизма неориентированных графов. - В кн.: Вычислительные системы. Вып.33. Новосибирск, 1969, с. 34-36.
6. CORNEIL D.G. Graph Isomorphism Technical Report, N 18, April, 1970, (2nd edition). University of Toronto Department of computer science.
7. HARARY F., WILCOX G. Boolean operations on graphs. - "Math. Scand.", 1967, v.20, N 1.
8. СКОРОВОГАТОВ В.В. Относительные разбиения и слои графов. - Настоящий сборник, с. 3-10.
9. ГИЗЕБУРТ А.Н., ЕМЕЛЬЯНОВ Э.Л. и др. Графопостроительная система "Вектор" и ее применения при машинном проектировании. - "Управляющие системы и машины", 1974, N 5, с. 96.
10. КОНИХОВСКИЙ А.С. Об одном алгоритме упорядочения массивов на ЭВМ. - В кн.: Математические вопросы управления производством. Вып. 4, 1972, с.69. (Моск. гос. ун-т).
11. ЗЫКОВ А.А. Теория конечных графов. Новосибирск, "Наука", СО АН СССР, 1969.

Поступила в ред.-изд.отд.  
25 июня 1976 года