Задание для курсовой работы

Цель курсовой работы: освоение математического аппарата задания, анализа графовых моделей дискретных систем и решения ряда основных задач на графах.

Индивидуальный вариант графов необходимо получить на сайте ws-dss.com (метод **graph_generator**).

Порядок выполнения:

1. Построить аналитически и графически объединение, пересечения, разность графов G (X, F) и H (Y, P), дополнение графа G по отображению до универсального.

2. Для графа $S = G \cup H$ построить матрицы смежности, инцидентности, достижимости; конденсацию графа. Определить все базовые и доминирующие множества, найти среди них минимальные.

3. Для графа $S = G \cup H$ построить Гамильтонов и Эйлеров путь (если они не существуют, то дополнить граф необходимыми дугами, обозначив их на графе). Гамильтонов путь построить двумя способами:

- с использованием алгоритма Фаулкса;
- с использованием алгоритма Робертса и Флореса.

Получить из графа S не ориентированный граф путем замены всех ориентированных дуг на неориентированные. Найти в полученном графе Эйлеров путь (если он не существует, то дополнить граф необходимыми дугами, обозначив их на графе).

Сравнить полученные результаты с результатами расчетов на ЭВМ (методы **hamiltonian_path** – для ориентированного графа и **eulerian_path** – *только для неориентированного графа*).

4. Определение связности графа.

4.1. Для заданного с помощью матрицы смежности неориентированного графа найти связные компоненты, используя алгоритм Фаулкса.

4.2. Представить заданный граф графически.

4.3. Найти связные компоненты в графе, используя метод **graph_connectivity** на сайте ws-dss.com. Сравнить полученные результаты.

5. Для рассмотренных задач:

- поиск Гамильтонова пути;
- поиск Эйлерова пути;
- определение связности графа;
- поиск базового и доминирующего множеств

предложить содержательное(предметное) описание.

6. Составить отчет. Ответить на контрольные вопросы.

Методические указания

При проектировании различных систем часто приходится решать задачи, которые можно назвать задачами согласования и упорядочения. Такие задачи, в частности, возникают при планировании работы цехов и предприятий, выпускающих широкую номенклатуру продукции с использованием в производственном процессе различных комбинаций станков и другого оборудования. Даже если рассматривается всего одна единица оборудования, на которой выполняются различные работы, задача определения порядка выполнения работ является часто довольно сложной. Аналогичные проблемы, возникают при определении очередности решения задач на ЭВМ или порядка доставки груза многим потребителям при наличии одной транспортной единицы; при определении наличия взаимосвязей между группами объектов в сложной информационной системе; при определении порядка регламентных проверок исправности всех каналов связи между звеньями какой-нибудь достаточно сложной системы, обслуживаемой одной ремонтной единицей; при определении подчиненности одних элементов системы другим, возможности передачи управляющих воздействий, определения совокупности пунктов управления, из которых возможна передача команд ко всем элементам сложной системы и т.д.

Решение многих из них эффективно сводятся к решению известных задач с использованием аппарата теории графов, в частности, таких как поиск гамильтонова пути в графе, определение связности графа, поиск эйлерового пути в графе, определение сильных компонент, базового и доминирующих множеств в графе и других.

Определение гамильтонова пути в графе

Гамильтонов путь в графе – это путь, проходящий через все вершины графа, причем каждая вершина встречается в нем только один раз. Если этот путь замкнут, то говорят о гамильтоновом цикле.

В качестве примера рассмотрим задачу определения очередности решения на ЭВМ вычислительной программы (алгоритма) А, состоящей из n подпрограмм A_i ; $i=1,2,\dots,n$, причем на порядок выполнения подпрограмм накладываются ограничения в виде

$$A_i \rightarrow A_j \ , \qquad (4.1)$$

т.е. выполнение подпрограммы A_i предшествует выполнению подпрограммы A_j , $i, j \in \{1,2,\dots,n\}$ или

$$A_i \leftrightarrow A_j \ , \qquad (4.2)$$

т.е. порядок выполнения программ A_i и A_j безразличен.

Эту задачу удобно геометрически отобразить в виде графа, где каждой i вершине соответствует A_i - подпрограмма, а связи между подпрограммами соответствуют дугам или ребрам. Причем дуга соответствует связи $A_i \rightarrow A_j$, а ребро - связи $A_i \leftrightarrow A_j$.

Если предполагается, что программа A будет выполнена, когда выполнены все подпрограммы A_i , i=1,2,…,n (без повторений), то определение очередности выполнения подпрограмм A_{i} сводится к определению на графе гамильтонова пути. Очевидно, что эту задачу можно решить перебором всех возможных вариантов ($n!$ в насыщенном графе) с одновременной проверкой выполнения всех условий (4.1), (4.2). При $n = 2$ максимально возможно всего два варианта, при $n = 3$ - шесть, при n=10 существует более трех с половиной миллионов возможных вариантов, из которых надо выбрать допустимые, удовлетворяющие условиям (4.1), (4.2).

Решение поставленной задачи может быть облегчено, если воспользоваться алгоритмом Фаулкса, позволяющим во многих случаях значительно уменьшить количество рассматриваемых возможных вариантов [1].

Алгоритм Фаулкса. Рассматривается n операций (вершины графа) $A_1, A_2, \ldots, A_n$, между которыми существуют соотношения:

$$A_i \rightarrow A_j, A_i \rightarrow A_j, \; i, j \in \{1,2,\ldots,n\} \qquad (4.3)$$

Задача состоит в том, что среди всех возможных путей найти, если это возможно, путь (или несколько путей), проходящий один и только один раз через каждую из вершин графа и удовлетворяющий написанным выше соотношениям (4.3), то есть гамильтонов путь.

Идея вычислений, которые необходимо выполнить при использовании алгоритма Фаулкса, заключается в нахождении матрицы R^N путем последовательного булевого умножения $\otimes$ матрицы R^1 достижимости не более чем за один шаг саму на себя N раз (матрица R^1 может быть получена из матрицы смежности R в результате подстановки единиц на главной диагонали).

$$R^N = \underbrace{R^1 \otimes R^1 \otimes \ldots \otimes R^1}_{N}$$

Элементы r^N_{ij} матрицы R^{N}**,** определяются как

$$r_{ij}^N = \bigvee_{k=1}^{n} r_{ik}^{N-1} \cdot r_{kj}$$

Вычисления следует прекратить, если $R^N = R^{N\text{-}1}$.

На каждом цикле вычисления возможны некоторые упрощения матрицы, связанные с выявлением «начальных» и «конечных» вершин, ограничивающих возможность дальнейшего построения ГП из общего множества возможных путей в графе.

Порядок решения задачи:

1. Задаем номер цикла вычислений N=1.

2. Используя соотношения (4.3), составляем граф, вершины которого есть операции, а направленные дуги графа определяются заданными соотношениями между операциями.

3. Составляем матрицу R^1 графа, элементы r^1_{ij} которой могут принимать значения 0 либо 1. Если r^1_{ij} = 1, то это указывает на то, что операция (вершина) A_j должна следовать за операцией (вершиной) A_i . Полученная таким образом матрица R^1 является булевой матрицей (каждый элемент может принимать значения 0 или 1). Она является исходной для всех дальнейших вычислений.

4. Используя матрицу R^N (на первом цикле вычислений $R^N = R^1$), для каждой вершины графа (операции) проверяем принадлежность ее к начальной или конечной вершине ГП по следующим правилам.

Вершина (операция) является *начальной* вершиной ГП, если во всей строке матрицы стоят единицы, а во всем столбце (за исключением их пересечения) - нули. Вершина (операция) является *конечной* вершиной ГП, то во всей строке матрицы стоят нули (за исключением их пересечения), а во всем столбце - единицы.

5. Упрощаем матрицу R^N графа, вычеркивая из нее строки и столбцы, соответствующие начальной или (и) конечной вершине ГП. Получаем матрицу $(R^N)'$.

6. Увеличиваем на единицу номер цикла, т.е. $N=N+1$.

7. Находим матрицу $(R^N)'$, выполнив булевое умножение матрицы $(R^{N-1})'$ на матрицу $(R^1)'$, которая получается из исходной матрицы R^1 вычеркиванием из нее строк и столбцов, соответствующих начальным и конечным вершинам ГП, найденным во всех предыдущих циклах вычислений. Булевое перемножение матриц производится по обычным правилам умножения, только вместо умножения используется конъюнкция, а вместо сложения – дизъюнкция.

8. Проверяем условие выполнения равенства матриц $(R^N)'$ и $(R^{N-1})'$.

Если это условие выполняется, то переходим к п. 10, Иначе - к п. 9.

9. Проверяем условие равенства всех элементов матрицы R^N единице. Если это условие выполняется, то нет необходимости в дальнейших вычислениях матрицы $(R^N)'$, так как очевидно, что $(R^{N+1})' = (R^N)'$, переходим к п. 10. Если нет - к п. 4.

10. Составляем матрицу R^b , возвращая в матрицу $(R^N)'$, строки и столбцы, вычеркнутые в п. 5 во всех циклах вычислений.

11. Составляем матрицу R^c , перегруппировывая одновременно строки и столбцы матрицы R^b таким образом, чтобы все нули были расположены под главной диагональю, а единицы - над ней.

Квадратные матрицы, состоящие из единиц, опирающихся на главную диагональ, образуют классы эквивалентности вершин (совокупности вершин графа, эквивалентных с точки зрения очередности их присутствия в гамильтоновом пути).

Если для данного графа мы получили m классов эквивалентности ($m \le n$), то есть $B_1,\ldots,B_m$, и в каждый класс эквивалентности B_d, $d = 1,2,\ldots,m$ входят S_d вершин графа, то можно сказать, что вершины, входящие в класс эквивалентности B_d, расположенные выше и левее класса эквивалентности B_l в матрице R^c будут предшествовать в Гамильтоновом пути вершинам из класса эквивалентности B_l .

Нахождение Гамильтоновых путей в этом случае значительно упрощается. В соответствии с классами эквивалентности мы получили упорядоченные группы вершин. В качестве начальной выбирается вершина из первого класса эквивалентности. Если их в нем несколько, то следующая выбирается из этого же класса и так до тех пор, пока не будут исчерпаны все вершины этого класса. Затем переходим к вершинам следующего класса и т.д. пока не будут использованы все вершины графа и построен ГП. Следует заметить, что при выборе очередной вершины необходимо обязательно проверять выполнение условий (4.3). Если условия не выполняются, очередность следования вершин из одного класса эквивалентности меняется. Максимальное число возможных вариантов сокращается до произведения $B_1! \cdot B_2! \cdot ,\ldots, \cdot B_m!$

Пример. Пусть программа A состоит из 6 операций (подпрограмм) $A_1, A_2,\ldots, A_6$, между которыми существуют следующие соотношения:

$$
\begin{array}{lllll}
A_1 \to A_2, & A_2 \to A_3, & A_3 \to A_4, & A_5 \to A_2, & A_6 \to A_2, \\
A_1 \to A_4, & A_2 \to A_4, & & A_5 \to A_4, & A_6 \to A_4, \\
A_1 \to A_6, & A_2 \to A_5, & & & A_6 \to A_5 \\
 & A_2 \to A_6 . & & &
\end{array}
\tag{4.4}
$$

Цель задачи состоит в нахождении пути, проходящего только один раз через все операции (вершины графа) и удовлетворяющего написанным соотношениям. Без учета ограничений (4.4) всего возможно 6! = 720 вариантов следования вершин.

Для решения задачи воспользуемся приведенным алгоритмом.

1. Зададим номер цикла $N = 1$.

2. Составим граф (рис. 4.1), соответствующий соотношениям (4.4).

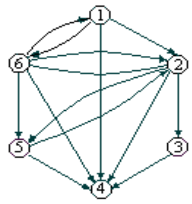


Рис. 4.1

3. Составим матрицу R^1 достижимости не более чем за один шаг.

$$R^1 = \begin{array}{c|cccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\ \hline A_1 & 1 & 1 & 0 & 1 & 0 & 1 \\ A_2 & 0 & 1 & 1 & 1 & 1 & 1 \\ A_3 & 0 & 0 & 1 & 1 & 0 & 0 \\ A_4 & 0 & 0 & 0 & 1 & 0 & 0 \\ A_5 & 0 & 1 & 0 & 1 & 1 & 0 \\ A_6 & 1 & 1 & 0 & 1 & 1 & 1 \end{array}$$

4. Проверим матрицу R^1 на наличие начальной или конечной вершин в графе.

Очевидно, что A_4 есть конечная вершина каждого гамильтонова пути, поскольку никакая дуга не имеет эту вершину своим началом, тогда как дуга, исходящая из любой другой вершины, достигает вершину A_4. Это свойство выражается наличием единиц во всем столбце A_4 и нулей во всей строке A_4 (за исключением их пересечения).

5. Вычеркиваем столбец и строку A_4. Получаем матрицу $(R^1)'$.

$$(R^1)' = \begin{array}{c|ccccc} & A_1 & A_2 & A_3 & A_5 & A_6 \\ \hline A_1 & 1 & 1 & 0 & 0 & 1 \\ A_2 & 0 & 1 & 1 & 1 & 1 \\ A_3 & 0 & 0 & 1 & 0 & 0 \\ A_5 & 0 & 1 & 0 & 1 & 0 \\ A_6 & 1 & 1 & 0 & 1 & 1 \end{array}$$

6. N=1+1=2.

7. Произведем булевое умножение матриц $(R^1)' \otimes (R^1)' = (R^2)'$.

Наличие 1 в матрице означает существование между соответствующими вершинами путей длиной меньшей или равной двум, а 0 - их отсутствие.

$$
(R^2)' = \begin{array}{c|ccccc}
 & A_1 & A_2 & A_3 & A_5 & A_6 \\
\hline
A_1 & 1 & 1 & 1 & 1 & 1 \\
A_2 & 1 & 1 & 1 & 1 & 1 \\
A_3 & 0 & 0 & 1 & 0 & 0 \\
A_5 & 0 & 1 & 1 & 1 & 1 \\
A_6 & 1 & 1 & 1 & 1 & 1
\end{array}
$$

8. Проверяем условие равенства $(R^2)'$ и $(R^1)'$. Так как они не равны, переходим к п. 9.

9. Проверяем условие равенства всех элементов матрицы единице. Так как оно не выполняется, то возвращаемся к п.4.

4. Рассмотрим матрицу $(R^2)'$.

Вершина A_3, является конечной вершиной гамильтонова пути на данном цикле вычислений.

5. Вычеркиваем столбец и строку A_3. Оставшаяся матрица имеет вид

$$
(R^2)' = \begin{array}{c|cccc}
 & A_1 & A_2 & A_5 & A_6 \\
\hline
A_1 & 1 & 1 & 1 & 1 \\
A_2 & 1 & 1 & 1 & 1 \\
A_5 & 0 & 1 & 1 & 1 \\
A_6 & 1 & 1 & 1 & 1
\end{array}.
$$

6. N=2+1=3.

7. Производим умножение $(R^2)' \otimes (R') = (R^3)'$.

$$
(R^3)' = \begin{array}{c|cccc}
 & A_1 & A_2 & A_5 & A_6 \\
\hline
A_1 & 1 & 1 & 1 & 1 \\
A_2 & 1 & 1 & 1 & 1 \\
A_5 & 1 & 1 & 1 & 1 \\
A_6 & 1 & 1 & 1 & 1
\end{array}.
$$

8. $(R^3)' \neq (R^2)'$

9. Все элементы $(R^N)'$, равны единице. Очевидно, что матрицы $(R^4)'$, $(R^5)'$ и т.д. также будут равны, т.к. будут иметь все элементы, равные единице.

Переходим к п.10.

10. Матрицу R^b получаем возвращением строк и столбцов, вычеркнутых в п. 5.

$$R^b = \begin{array}{c|cccccc} & A_1 & A_2 & A_3 & A_4 & A_5 & A_6 \\ \hline A_1 & 1 & 1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & 1 & 1 & 1 \\ A_3 & 0 & 0 & 1 & 1 & 0 & 0 \\ A_4 & 0 & 0 & 0 & 1 & 0 & 0 \\ A_5 & 1 & 1 & 1 & 1 & 1 & 1 \\ A_6 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}$$

11. Перегруппируем строки и столбцы матрицы R^{b} так, чтобы все нули были расположены под главной диагональю, а единицы - над ней: Получаем матрицу R^{c}

$$R^c = \begin{array}{c|cccc|cc} & A_1 & A_2 & A_5 & A_6 & A_3 & A_4 \\ \hline A_1 & 1 & 1 & 1 & 1 & 1 & 1 \\ A_2 & 1 & 1 & 1 & 1 & 1 & 1 \\ A_5 & 1 & 1 & 1 & 1 & 1 & 1 \\ A_6 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline A_3 & 0 & 0 & 0 & 0 & \boxed{1} & 1 \\ A_4 & 0 & 0 & 0 & 0 & 0 & \boxed{1} \end{array}$$

Таким образом, получили три класса эквивалентности:,

- в первый класс входят вершины A_1, A_2, A_5, A_6;
- во второй - вершина A_3;
- в третий - вершина A_4.

Т.е. в гамильтоновом пути вершины A_1, A_2, A_5, A_6 предшествуют вершине A_3, которая, в свою очередь, предшествует вершине A_4. Всего возможно $4! \cdot 1! \cdot 1! = 24$ вариантов следования вершин.

С учетом необходимости выполнения условий (4.4) определение гамильтонова пути становится достаточно простым делом. В данном примере получим единственный ГП: $A_1, A_6, A_5, A_2, A_3, A_4$.

Алгоритм Фаулкса в общем случае не решает однозначно задачу определения Гамильтонова пути, он только частично упорядочивает множество вершин, снижая тем самым размерность решаемой задачи. В случае, когда в каждый класс эквивалентности входит только одна вершина, алгоритм Фаулкса определяет гамильтонов путь однозначно.

В настоящее время нет критерия (простого), который позволял бы ответить на вопрос о существовании Гамильтонова пути в графе G. Существуют различные по трудоемкости методы поиска такого пути в графе, которые отвечают на вопрос существования такого пути только найдя его.

Например: **Метод перебора** (первоначально предложен американскими учеными **Робертсом и Флоресом**). Он основан на последовательном формировании элементарного пути до тех пор, пока либо получится Гамильтонов путь, либо не станет ясно, что формируемый путь к Гамильтонову не приводит. Тогда этот путь пытаются скорректировать таким образом, что либо будет найден Гамильтонов путь, либо исчерпаны все возможности по его корректировке. Рассмотрим метод более подробно.

Пусть задан граф G=(X,F). Необходимо сформировать множество S вершин, образующих Гамильтонов путь. Построим матрицу M – задающую множество вершин графа G, достижимых непосредственно из каждой его вершины. Число строк в ней равно числу элементов множества X, а число столбцов – максимальной из полустепени исхода по всем вершинам графа.

При построении элементарного пути в качестве отправной примем некоторую вершину (например, $x_1 \in X$) и поместим ее в множество S. $S=\{x_1\}$

К выбранной вершине x_1 последовательно добавляется первая из вершин, указанных в соответствующей вершине x_1 строке матрицы M (например, $x_к$). $S=\{x_1,x_к\}$

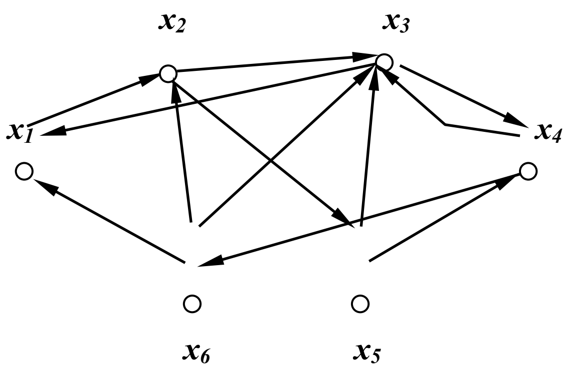
Затем к ним добавляется первая из вершин, расположенных в строке $x_к$ матрицы M и т.д. Причем каждая «новая» вершина не должна ранее принадлежать множеству S. Если она ему уже принадлежит, то берется следующая вершина из этой же строки. Если такой вершины в строке $x_к$ нет, то необходимо вернуться на предыдущий шаг и вместо $x_к$ выбрать следующую вершину, указанную в данной строке (например, x_l). При этом множество $S=\{x_1,x_l\}$.

Если это удается, то переходим к строке x_l и двигаемся дальше до тех пор, пока число элементов в множестве S не будет равно n (Гамильтонов путь). $S=\{x_1,x_l,\ldots.\}$

Если других гамильтоновых путей искать не требуется, то задача решена.

В противном случае возвращаемся на предыдущий шаг и пытаемся выбрать другое продолжение пути и т.д. до тех пор, пока не все гамильтоновы пути, идущие из вершины x_1 (если они существуют) будут найдены и процедура возвращения назад не сделает множества S пустым. (При возвращении на предыдущий шаг последний элемент из множества S исключается)

Пример:



Матрица M^T будет иметь вид:

x_1	x_2	x_3	x_4	x_5	x_6
x_2	x_3	x_1	x_3	x_3	x_1
-	x_5	x_4	x_6	x_4	x_2
-	-	-	-	-	x_3

В качестве начальной выбираем вершину x_1

1. $S=\{x_1\}$ К ней добавляем вершину x_2 из строки, соответствующей вершине x_1
2. $S=\{x_1,x_2\}$ Добавляем x_3
3. $S=\{x_1,x_2,x_3\}$ Добавляем x_4
4. $S=\{x_1,x_2,x_3,x_4\}$ Добавляем x_6
5. $S=\{x_1,x_2,x_3,x_4,x_6\}$ Дальше добавлять из строки, соответствующей вершине x_6 ничего нельзя, т.к. все ее элементы уже входят в S. Возвращаемся на предыдущий шаг
6. $S=\{x_1,x_2,x_3,x_4\}$ Кроме x_6 добавить нечего. Возвращаемся
7. $S=\{x_1,x_2,x_3\}$ Кроме x_4 добавить нечего. Возвращаемся
8. $S=\{x_1,x_2\}$ Добавляем x_5
9. $S=\{x_1,x_2,x_5\}$ Добавляем x_3
10. $S=\{x_1,x_2,x_5,x_3\}$ Добавляем x_4
11. $S=\{x_1,x_2,x_5,x_3,x_4\}$ Добавляем x_6
12. $S=\{x_1,x_2,x_5,x_3,x_4,x_6\}$ Гамильтонов путь.

Дуга (x_6,x_1) дает Гамильтонов цикл. Для поиска других гамильтоновых путей – возвращение на предыдущий шаг.

13. $S=\{x_1,x_2,x_5,x_3,x_4\}$ Возвращение
14. $S=\{x_1,x_2,x_5,x_3\}$ Возвращение
15. $S=\{x_1,x_2,x_5\}$ Добавляем x_4
16. $S=\{x_1,x_2,x_5,x_4\}$ Добавляем x_6
17. $S=\{x_1,x_2,x_5,x_4,x_6\}$ Добавляем x_3
18. $S=\{x_1,x_2,x_5,x_4,x_6,x_3\}$ Гамильтонов путь.

Дуга (x_3,x_1) дает Гамильтонов цикл. Для поиска других гамильтоновых путей – возвращение на предыдущий шаг.

$S=\{x_1,x_2,x_5,x_4,x_6\} \Rightarrow \{x_1,x_2,x_5,x_4\} \Rightarrow \{x_1,x_2,x_5\} \Rightarrow \{x_1,x_2,\} \Rightarrow \{x_1\} \Rightarrow S=\varnothing$

Таким образом, в графе существует два гамильтоновых пути, идущих из вершины a.

Указанный алгоритм прост. Он может быть модернизирован. Легко реализуется на ЭВМ.

Определение связности графа

Неориентированный граф называется **связным**, если для любой пары вершин $x_i,x_j \in X$ существует хотя бы один маршрут, их соединяющий. В противном случае граф несвязан. Несвязный граф может быть единственным образом разбит на группы взаимосвязанных вершин (на **связные компоненты**) таким образом, что не существует маршрута, соединяющей какие-либо из вершин различных компонент.

Решим задачу определения связности графа с помощью элементов алгоритма Фаулкса.

Пусть дана матрица смежности R неориентированного графа G (матрица R симметрична относительно главной диагонали), имеющего n вершин. Необходимо определить, является ли этот граф связным. В случае, если граф несвязный, то требуется выделить из него все m связных подграфов $G_1, G_2,\ldots, G_m$ таких, что, например, из любой вершины подграфа G_1 можно найти путь или маршрут, ведущий к любой другой вершине этого же подграфа, но нельзя найти путь или маршрут ни к одной вершине, не принадлежащей данному подграфу G_1. Требуется определить, какие вершины входят в каждый из подграфов.

На основании матрицы смежности R получим матрицу достижимости не более чем за один шаг R^1.

Затем, в соответствии с алгоритмом Фаулкса, найдем матрицу R^N путем булевского перемножения матриц ($R^N = R^{N-1} \otimes R^1$) до тех пор, пока не достигнем $R^N = R^{N-1}$. Полученная матрица R^N является матрицей достижимости графа G. Она позволяет выявить связные подграфы и номера вершин, входящих в связные подграфы. Для этого необходимо в некоторой i - строке матрицы R^N найти элементы $r^N_{ij} = 1$, $i, j = 1,\ldots,n$. Номера столбцов j, где $r_{ij} = 1$, и будут номерами вершин, входящих в связный подграф, включающий в себя i-ю вершину. Рассматривая последовательно все несовпадающие строки матрицы можно выявить все m связных подграфов исходного графа G.

В случае ориентированного графа, для решения задачи выделения связных подграфов, не связных между собой путем или маршрутом, необходимо перейти от ориентированного графа к неориентированному, заменив дуги неориентированными ребрами (звеньями). При возникновении параллельных звеньев оставить для дальнейшего рассмотрения только по одному звену, связывающему соответствующие вершины. В остальном процесс поиска связных подграфов не меняется.

Пример.

Пусть задана матрица R^1 достижимости не более чем за один шаг неориентированного графа G, имеющего $n = 8$ вершин.

$$R^1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Необходимо проверить граф на связность и выявить все связные подграфы.

1. Зададим начальный номер цикла вычислений: $N = 1$.

2. $N = N + 1 = 2$.

3. Производим умножение $R^1 \otimes R^1 = R^2$

$$R^2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

4. Проверяем условие равенства R^2 и R^1. Они не равны. Переходим к п. 2.

2) $N = 2 + 1 = 3$.

3) Умножение $R^2 \otimes R^1 = R^3$.

$$R^3 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

4) Проверяем условие равенства матриц R^2 и R^3. Так как они равны, то это означает, что путь длиной 2 - максимальный в данной графе, и дальнейшее перемножение матрицы не имеет смысла, так как мы будем получать одну и ту же матрицу. Найденная матрица R^3 есть матрица достижимости исходного графа. Наличие нулей в полученной матрице показывает, что между соответствующими элементами отсутствует путь любой длины и, следовательно, данный граф — несвязный.

5. Выбираем начальную вершину для выявления связных подграфов: $i = 1$.

6. В строке $i = 1$ выбираем все элементы r^3_{1j}, равные единице ($j=1,\ldots,n$). $r^3_{11} = r^3_{12} = r^3_{17} = r^3_{18} = 1$. Номера столбцов, в которых $r^3_{ij} = 1$ - это номера вершин первого связного подграфа, входящего в данный граф. То есть вершины 1, 2, 7, 8 образуют связный подграф G_1.

7. Определяем номер очередной вершины $i = 1 + 1 = 2$. Так как $2 \le 8$, то на 8 (иначе на 9).

8. Проверим – входит ли данная вершина в найденные связные подграфы. Если среди вершин найденных связных подграфов есть данная вершина, то на 7, иначе на 6.

Так как среди вершин связного подграфа G_1 есть вершина $j=2$, то возвращаемся к п.7.

7) $i = 2 + 1 = 3$. Так как $3 \le 8$, то на 8.

8) Среди вершин связного графа G_1 нет вершины $j=3$. Возвращаемся к п. 6.

6.) В строке 3 выбираем элементы, равные единице: $r^3_{33} = r^3_{36} = 1$. Следовательно, во второй связный подграф G_2 входят вершины 3, 6.

7) Определяем номер очередной вершины $i = 3 + 1 = 4$. Так как $4 \le 8$, то на 8.

8.) Среди вершин связных подграфов G_1 и G_2 нет вершины $j = 4$. Возвращаемся к п. 6.

6.) В строке 4 выбираем элементы, равные единице. $r^3_{44} = r^3_{45} = 1$. Следовательно, в третий связный подграф G_3 входят вершины 4, 5.

7) $i = 4 + 1 = 5$. Так как $4 \le 8$, то на 8.

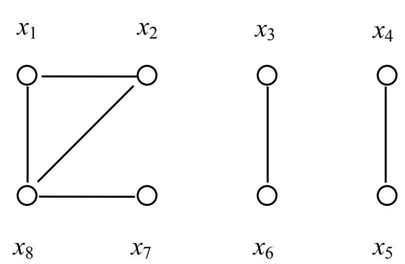
8) Среди вершин связных графов G_1, G_2, G_3 есть вершина $j = 5$. Переходим к п.7.

Аналогично проверяются вершины $j = 6, 7, 8$. Они входят в уже полученные выше подграфы.

9. Если проверены все вершины, то все связные подграфы найдены. Конец.

Таким образом в результате работы алгоритма найдены три связных подграфа:

$G_1 = \{1, 2, 7, 8\}$; $G_2 = \{3, 6\}$; $G_3 = \{4, 5\}$ (см. рисунок 4.2)



.

Рис 4. 2.

Определение эйлерового пути в графе

Одна из важных задач теории графов заключается в нахождении пути (маршрута), содержащего по одному разу все ребра графа. Этот путь называется **эйлеровым путем (ЭП)** или эйлеровой линией в графе, а задачу его нахождения называют задачей Эйлера. Графы, на которых существуют замкнутые эйлеровы линии, называют **эйлеровыми графами**.

На практике задача нахождения ЭП на графе может встретиться, например, при определении порядка регламентных проверок исправности каналов связи между звеньями какой-нибудь достаточно сложной системы, обслуживаемой одной ремонтной единицей.

Ответ на вопрос о существовании ЭП на графах дают теоремы Эйлера.

Теорема Эйлера для неориентированных графов: для того чтобы на графе существовал замкнутый эйлеров путь (маршрут), необходимо и достаточно, чтобы граф был связным и степени всех его вершин были четными [1, 8].

Теорема Эйлера для ориентированных графов: для того чтобы на ориентированном графе существовал замкнутый ЭП, необходимо и достаточно, чтобы граф был связным и для каждой его вершины полустепень исхода равнялась полустепени захода [1, 8].

Для нахождения ЭП на графе (если он существует - см. теорему Эйлера) удобно воспользоваться следующим алгоритмом (для неориентированного графа):

0. Задается начальный шаг итерации k =0. Задается начальная (она же и конечная в случае замкнутого ЭП) вершина ЭП V_0. Если, например, $V_0 = A_3$, то это означает, что начальной вершиной ЭП на шаге k = 0 выбрана третья вершина графа. Все ребра графа считаются непомеченными. Число непомеченных ребер S равно числу ребер графа.

1. k=k+1. Определяются все вершины, непосредственно связанные непомеченными ребрами с вершиной V_{k-1}.

2. Определяется число S_k непомеченных ребер, инцидентных вершине V_{k-1}.

Если S_k= 1, то вторая вершина, инцидентная этому ребру, становится очередной вершиной V_k ЭП. Это ребро помечается. Возвращаемся на п. 1.

Если S_k >1, то среди вершин, смежных V_{k-1}, выбирается вершина с наименьшим номером k, которая становится очередной вершиной V_k ЭП. Ребро, соединяющее эту вершину с вершиной V_{k-1}, помечается. Возвращаемся на п. 1.

3. S_k = 0. Для вершины, принадлежащей полученному таким образом пути и имеющей непомеченные инцидентные ребра, строится цикл по непомеченным ребрам. Определение новой вершины этого цикла начинается с п. 1.

4. Построенный таким образом новый цикл объединяется с ранее построенным в вершине, с которой начиналось построение нового цикла.

5. Если в цикле есть еще вершины, имеющие непомеченные инцидентные ребра, то возвращаемся на п. 3. Если таких ребер нет, то ЭП найден.

Отметим, что выбор вершины с наименьшим номером нужен для того, чтобы была определенность при выборе вершины на каждом шаге. Можно, однако, предложить выбор вершин и с наибольшим номером.

В качестве примера рассмотрим задачу нахождения ЭП для графа (рис. 4.3). ЭП существует, так как степени всех вершин четные.

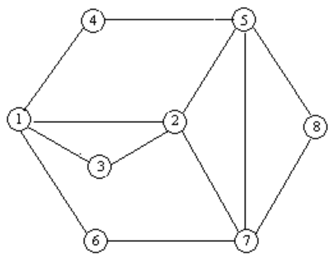


Рис. 4.3

0. k = 0. Выберем начальную вершину с номером 3, т.е. V_0 = 3. Число непомеченных ребер S = 12.

1. k = 1. С вершиной V_0 = 3 связаны вершины 1, 2.

2. Число непомеченных ребер для 3-й вершины S_1 = 2. Следующей вершиной ЭП является вершина с номером 1. Ребро 3-1 помечается. Возвращаемся к п. 1.

1. k = 2. С вершиной 1 связаны вершины 2, 4, 6.

2. S_2 = 3. Вершина 2. Помечается ребро 1-2. Возвращаемся к п. 1.

1. k = 3. С вершиной 2 связаны вершины 3, 5, 7.

2. S_3 = 3. Вершина 3. Помечается ребро 2-3. Возвращаемся к п. 1.

1. k = 4. С вершиной 3 связаны вершины 1, 2.

2. S_4 = 0. Вершин таких нет. Следовательно, получен цикл 3-1-2-3, который еще не является ЭП.

3. Вершины 1 и 2, входящие в этот цикл, имеют непомеченные инцидентные ребра. Рассмотрим вершину 1 в качестве начальной точки следующего цикла A_4 = 1.

1. k = 5. Вершины 4, 6.

2. S_5 = 2. Вершина 4. Ребро 1-4.

1. k = 6. Вершина 5.

2. S_6 = 1. Вершина 5. Ребро 4-5.

1. k = 7. Вершины 2, 7, 8.

2. $S_7 = 3$. Вершина 2. Ребро 5-2.

1. $k = 8$. Вершина 7.

2. $S_8 = 1$. Вершина 7. Ребро 2-7.

1. $k = 9$. Вершины 5, 6, 8.

2. $S_9 = 3$. Вершина 5. Ребро 7-5.

1. $k = 10$. Вершина 8.

2. $S_{10} = 1$. Вершина 8. Ребро 5-8.

1. $k = 11$. Вершина 7.

2. $S_{11} = 1$. Вершина 7. Ребро 8-7.

1. $k = 12$. Вершина 6.

2. $S_{12} = 1$. Вершина 6. Ребро 7-6.

1. $k = 13$. Вершина 1.

2. $S_{13} = 1$. Вершина 1. Ребро 6-1.

1. $k = 14$. Вершин, связанных непомеченными ребрами, нет.

3. $S_{14} = 0$. В найденных циклах нет вершин, имеющих инцидентные непомеченные ребра.

4. Для нахождения ЭП необходимо объединить два найденных цикла 3-1-2-3 и 1-4-5-2-7-5-8-7-6-1 в вершине 1. Результатом объединения будет путь: 3-1-4-5-2-7-5-8-7-6-1-2-3.

5. Так как в этом пути нет вершин с непомеченными ребрами, то найденный путь является эйлеровым.

Нахождение сильных компонент, базового и доминирующего множеств

При определении сильных компонент и базового множества ориентированного графа G используется **матрица достижимости** L и **контрдостижимости** Q графа [6]. Элементы l_{ik} и q_{ik} этих матриц определяются следующим образом:

$$l_{ik} = \begin{cases} 1 - \text{если вершина } x_k \text{ достижима из вершины } x_i; \\ 0 - \text{если вершина } x_k \text{ не достижима из вершины } x_i; \end{cases}$$

$$q_{ik} = \begin{cases} 1 - \text{если вершина } x_i \text{ достижима из вершины } x_k; \\ 0 - \text{если вершина } x_i \text{ не достижима из вершины } x_k; \end{cases}$$

<u>Матрица достижимости L</u> графа G, имеющего n вершин, может быть получена путем последовательного булевого умножения матрицы R^1 графа (достижимости не более чем за один шаг) саму на себя N раз.

$$R^N = \underbrace{R^1 \otimes R^1 \otimes \ldots \otimes R^1}_{N}$$

Значение N определяется из условия, при котором $R^N = R^{N-1}$, причем $N \le n$-1.

Матрица контрдостижимости Q графа G может быть найдена из матрицы достижимости путем ее транспонирования, то есть $Q = L^{\mathrm{T}}$.

Пример.

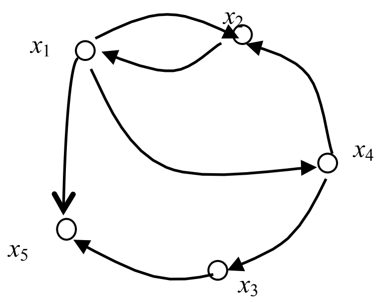


Рис. 4.4

$$R^1 = \begin{pmatrix} 11011 \\ 11000 \\ 00101 \\ 00110 \\ 00001 \end{pmatrix}, \qquad L = \begin{pmatrix} 11111 \\ 11111 \\ 00101 \\ 11111 \\ 00001 \end{pmatrix}, \quad Q = \begin{pmatrix} 11010 \\ 11010 \\ 11110 \\ 11010 \\ 11111 \end{pmatrix}.$$

Под **сильной компонентой** (**СК**) графа G будем понимать подграф G' графа G максимальной размерности (максимальное число вершин), являющийся сильно связным графом, который не содержится в любом другом сильном подграфе графа G. Ориентированный граф $G=(X,F)$ является сильно связным, если для любой пары вершин (x_i, x_j) существует хотя бы один путь как из $x_i \to x_j$, так и наоборот из $x_j \to x_i$.

Сильные компоненты произвольного графа G могут быть найдены в результате поэлементного логического перемножения матриц достижимости и контрдостижимости. Номера столбцов в i-й строке, для которых элемент матрицы равен 1, соответствуют номерам вершин, входящих в данную сильную компоненту. В нашем случае получим

$$L \cap Q = \begin{pmatrix} 11010 \\ 11010 \\ 00100 \\ 11010 \\ 00001 \end{pmatrix}.$$

Для рассмотренного выше графа (рис. 4.4) получим три сильные компоненты: $CK_{x1}=\{x_1, x_2, x_4\}$, $CK_{x2}=\{x_3\}$, $CK_{x3}=\{x_5\}$.

На сильных компонентах, как на вершинах, может быть построен новый граф $G^*=(X^*, F^*)$ – который называется **конденсацией** графа G. Каждая его вершина отображает множество вершин, соответствующих одной из сильных компонент исходного графа G, а дуга (x_{ξ^*}, x_{η^*}) существует в G^* тогда и только тогда, когда в графе G существует дуга (x_i, x_j) такая, что $x_i \in СКx_{\xi}$, а $x_j \in СКx_{\eta}$.

Для исходного графа конденсацией будет граф, состоящий из 3-х вершин (рис. 4.5).

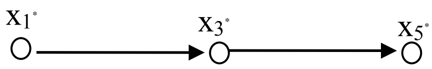


Рис. 4.5.

Базовым множеством графа G называется такое множество B вершин этого графа, из которых достижима любая вершина графа и не существует его подмножества, обладающего таким же свойством достижимости.

Доминирующим множеством вершин $S \subseteq X$ графа $G=(X,F)$ называется такое множество вершин, что для каждой вершины $x_j \notin S$ существует дуга (путь длиной 1), идущая от некоторой вершины множества S к данной вершине x_j и не существует его подмножества, обладающего таким же свойством.

Задача нахождения базового и доминирующего множеств в графе сводится к определению такого множества вершин, из которых достижима любая вершина графа (но за различное число шагов). Для этого целесообразно использовать матрицы достижимости и смежности соответственно.

Задача определения минимального доминирующего множества эквивалентна задаче нахождения минимальной ДНФ на импликантной матрице (см. работу №2 настоящего руководства), т.е. такого наименьшего множества строк, что каждый столбец содержит единицу, хотя бы в одной из этого множества строк. Поиск минимального доминирующего множества осуществляется на матрице R^1 достижимости не более чем за один шаг. Таким минимальным доминирующим множеством для графа на рис. 4.4 является множество, состоящее из вершин x_1, x_3 или x_1, x_4 .

Базовое множество определяется аналогично. Но, при этом, в отличие от определения доминирующего множества, задача решается не на матрице R^1, а на матрице достижимости L.

В случае графа большой размерности для поиска базового множества целесообразно сначала найти конденсацию графа, затем базу конденсации, и только затем базу исходного графа, выбирая по одной вершине из каждой сильной компоненты, входящей в базу конденсации. Размерность задачи в этом случае существенно снижается.

Для графа на рис. 4.4 базовым множеством будут вершины или x_1, или x_2, или x_4, (одна из вершин, входящих в сильную компоненту $CK_{x1}=\{x_1, x_2, x_4\}$, соответствующую базе конденсации графа G^* – вершине x_{1^*}).

Для нахождения базовых и доминирующих множеств могут использоваться алгоритмы нахождения ТДНФ и МДНФ из лабораторной работы №3.

Контрольные вопросы

1. Что такое Гамильтонов путь? Как определить по матрице смежности начальные и конечные вершины Гамильтонова пути в графе?
2. Что такое связный граф? Как найти связные подграфы в неориентированном графе.
3. Что такое Эйлеров путь? Во всяком ли связном графе существует ЭП?
4. Как по матрице смежности неориентированного графа определить существование ЭП?
5. Что такое сильная компонента графа?
6. Что такое базовое и доминирующее множества графа?
7. Каковы свойства сильной компоненты, минимального базового и доминирующего множеств?
8. Как находится матрица достижимости и контрдостижимости?
9. Что такое конденсация графа.
10. В чем сходство и различие алгоритмов определения базового и доминирующего множеств?

ЛИТЕРАТУРА

1. Босин П.Л., Булыгии В.С., Кулешов К.Г. Лабораторные работы но курсу «Основы теории конечных динамических систем», — М.: МАИ, 1985.

2. Булыгин В.С. Логические основы теории дискретных устройств: учеб. пособие. - М.: МАИ, 1983.

3. Булыгин В.С. Основы проектирования дискретных устройств АСУ: учеб. пособие. - М.: МАИ, 1982.

4. Поспелов Д.А. Логические методы анализа и синтеза схем. - М.: Энергия, 1968.

5. Просветов Г.И. Дискретная математика: задачи и решения: учебное пособие. – М.: БИНОМ. Лаборатория знаний,2008. – 222 с.

6. Кристофидес Н, Теория графов: Алгоритмический подход. - М.: Мир, 1978.

7. Булыгин В.С., Ескин В.И.. Модели дискретных устройств с памятью в АСУ: Учеб. пособие. - М.: Изд-во МАИ, 1993. - 72 с.: ил.

8. Мелихов А.Н. Ориентированные графы и конечные автоматы. - М.: Наука, 1971.

9. Булыгин В.С., Ескин В.И. Лабораторные работы «Дискретная математика (логические функции, конечные автоматы, графы)», — М.: МАИ, 2011.