Serdica
Bulgariacae mathematicae publicationes

Сердика
Българско математическо списание

ПОИСК ИНФОРМАЦИИ ПУТЕМ СИНТЕЗА СЛОЖНЫХ ВЫСКАЗЫВАНИЙ

ГЕОРГИ Д. КАЛИНОВ, ХРИСТО З. ЗЛАТАНОВ, ТОДОР Н. ТОДОРОВ

Рассматривается информационно-логическая задача поиска в информационном массиве, представленном в виде графа. Задача сводится к перечислению всех подграфов, изоморфных данному графу-запросу.

1. Введение. В практике автоматизированных информационных систем (АИС) приходится решать информационно-логические задачи, рассматривающие информационный массив, содержащий высказывания о свойствах двух категорий объектов — категории „вершина“ и категории „дуга“. Вопросы к системе могут быть представлены в виде графа-запроса, вершины и дуги которого — высказывания, аналогичны высказываниям в основном информационном массиве. В сущности, необходимо найти в основном графе все подграфы, изоморфные графу-запросу. В таком случае запрос — это сложное высказывание, которое в явной форме не содержится в основном массиве. АИС, использующая предположенный алгоритм для синтеза сложных высказываний, выводимых из основного массива и изоморфных орграфу-запросу, может выдавать в ответ на запрос:

— все выводимые высказывания;

— первые M выводимые высказывания;

— первые M выводимые высказывания и число всех выводимых высказываний.

2. Описание алгоритма. Сначала введем некоторые обозначения. Если A — вершина, то $\breve{A}$ — множество вершин, из которых выходят входящие в A дуги, $\hat{A}$ — множество вершин, в которые входят выходящие из A дуги, а $\bar{A}=\breve{A}\cap\hat{A}$. Через $|\breve{A}|$ и $|\hat{A}|$ обозначим число элементов $\breve{A}$ и $\hat{A}$ и пусть $N(A)=|\breve{A}|+|\hat{A}|$.

Будем предполагать, что запрос представляет собой связный граф. В другом случае граф-запрос можно рассматривать как два или больше запросов.

Перед тем как начать поиск, вершины, участвующие в запросе, упорядочиваются следующим способом:

— первой выбирается одна из вершин с самым большим N;

— следующей, из числа остальных, выбирается та вершина, которая связана с вершиной, выбранной как можно раньше, и имеет максимальное число связей с остальными выбранными вершинами (предположение, что граф-запрос связан, обуславливает существование такой вершины); при получении нескольких таких вершин выбирается вершина с максимальным N.

После того как запрос будет упорядочен указанным способом, осуществляется поиск в информационном массиве.

СЕРДИКА Българско математическо списание. Том 10, 1984, с. 248—252.

Для иллюстрации алгоритма поиска рассмотрим пример нахождения в информационном массиве всех подграфов, изоморфных графу-запросу, представленный на рис. 1. В примере A является высказывание о некоторой личности, а (A, B)—высказывание о наличии определенного отношения между соответствующими лицами и т. д.

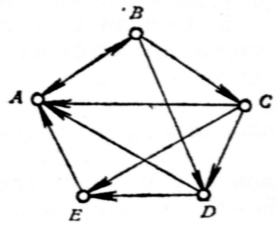


Рис. 1

На рис. 2 дана блок-схема алгоритма, который будет формироваться автоматично, для поиска указанного графа-запроса. Каждый из операторов в блок-схеме (например, $B:|\breve{B}|\geqq 1$, $|\hat{B}|\geqq 3$, $B\in\overline{A}$) фиксирует вершину (B), удовлетворяющую конъюнкцию из условий, указанных для этой вершины. Притом эта вершина отличается от вершин, фиксированных предыдущими операторами. При вхождении в оператор „сверху" берется та вершина (из остальных инцидентных вершин), для которой в наибольшей степени выполняются неравенства, участвующие в условиях (это имеет значение в случае, когда необходимо найти только определенное число ответов на за-

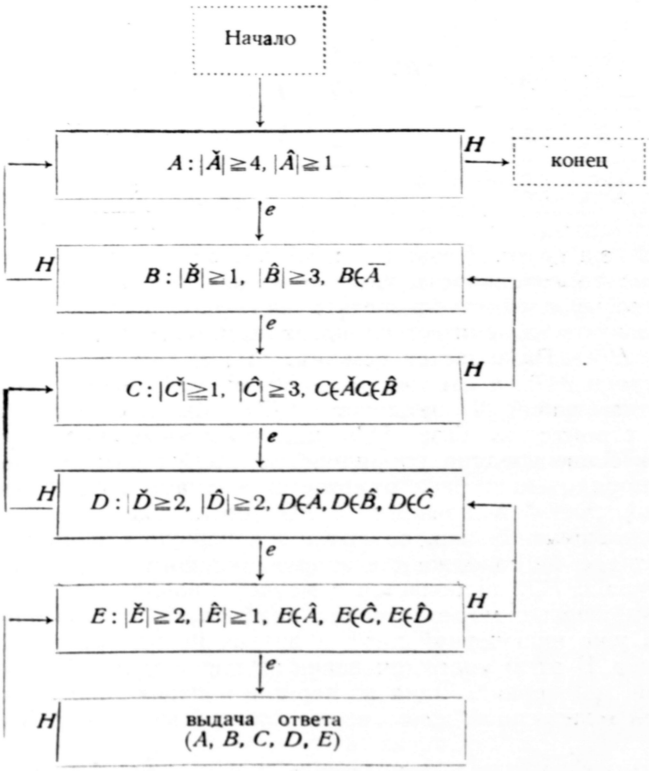


Рис. 2

прос). При каждом вхождении в оператор „снизу“ фиксируется вершина, которая до этого еще не была фиксирована (в блок-схеме этот выход обозначается через *e* — есть) или выдается сообщение, что нет возможности найти такую вершину (*H* — нет).

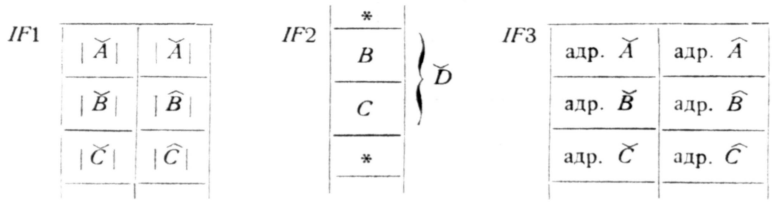
Описанный алгоритм экспериментирован на машине типа ЕС.

3. Описание реализации с помощью языка PL/1. 3.1 Информационный массив. Машинное представление информационного массива состоит из следующих трех компонентов:

— массив *IF*1, содержащий характеристики $|\breve{A}|$ и $|\hat{A}|$, для любого A, принадлежащего информационному массиву;

— массив *IF*2, содержащий группы вершин, принадлежащих информационному массиву, где каждой группе поставлена в соответствие вершина, смежная каждой вершине группы, и все дуги, инцидентные данной вершине и вершинам из соответствующей группы, являются только исходящими или только входящими для данной вершины;

— массив *IF*3, содержащий информацию о начальном адресе каждой группы. Вершины информационного массива пронумерованы и входят явно только в *FI*2. Другие два массива упорядочены последовательно по номеру вершины.



3.2 Граф-запрос. Связный граф-запрос предоставляется алгоритму поиска как последовательность упорядоченных пар вершин. Алгоритм преобразовывает эту последовательность в виде двух массивов, которые в дальнейшем используют для синтеза сложных высказываний. Первый массив (*FG*1) аналогичный *IF*1. Различается тем, что имена вершин графа-запроса присутствуют явно в *FG*1, и что он упорядочен так, чтобы ускорить синтез сложных высказываний. Упорядочение *FG*1 было описано в т. 2. Второй массив (*FG*2) строится на базе *FG*1 и последовательности упорядоченных пар. В этом массиве введено упорядочение следующим образом.

Любая упорядоченная пара отмечается метками „ввод“ или „проверка“. Пары с меткой „ввод“ включаются в *FG*2 последовательно по именам вершин в *FG*1, начиная с пар, содержащих первую вершину из *FG*1. После этого следуют все — содержащие вторую вершину из *EG*1 и т. д.

Первая пара в *FG*2 включается с меткой „ввод“. Для включения каждой следующей пары с меткой „ввод“ необходимо, чтобы одна ее вершина принадлежала уже включенной паре, а другая не принадлежала ни одной из включенных пар. В этом упорядочивании за парой с меткой „ввод“ следуют пары с меткой „проверка“. Одна из вершин в парах „проверка“ принадлежит последней включенной паре „ввод“, а другая—одной из включенных уже пар.

Одну и ту же пару нельзя включить и как „ввод“, и как „проверка“. Пара, для которой в описанном процессе возникает возможность включения как „проверка“, обязательно включается с этой меткой.

При построении $FG2$ указанные выше условия обязательны. Они необходимы для ускорения работы алгоритма синтеза сложных высказываний.

3.3 Алгоритм синтеза. Основная задача алгоритма синтеза состоит в заполнении матрицы $M(I, K)$ размерностью $m \times 3$, где m — число вершин в $FG1$. Отдельные столбцы матрицы содержат следующие элементы:

а. В первом столбце — вершины из графа-запроса.

б. Во втором столбце — вершины из информационного массива.

в. В третьем столбце — элементы из $IF3$ или элементы из $IE3$, над которым осуществлены операции типа $(IF3(I)) = (IF3(I) + 1$. Каждый элемент показывает позицию в $IF2$, соответствующей вершине второго столбца. $(IF3(I)$ — содержимое в $IF3(I)$.

Блок-схема (рис. 3) иллюстрирует работу программы.

Поясним некоторые из использованных далее понятий. Когда актуальная $FG2(L)$ имеет метку „ввод“, алгоритм выполняет включение вершины. Это означает, что вершина в $FG2(L)$, которая первый раз встречается алгоритмом, записывается в $M(I, 1)$. „Невозможность“ $FG2$ (L) в данном случае означает, что алгоритм не может найти элемент, который должен поставить в $M(I, 2)$.

Когда актуальная $FG2(L)$ имеет метку „проверка“ и имеют место равенства

$$(FG2(L)) \equiv \text{проверка}: (A, B),$$

$$(M(I, 1)) \equiv A \quad \text{и}$$

$$(M(R, 1)) \equiv B,$$

алгоритм проверяет существование упорядоченной пары $((M(I, 2)), (M(R, 2)))$ в основном масиве. Если такая пара не существует для $FG2(L)$, возникает условие „невозможность“. Если для пары с меткой „проверка“ возникает условие "невозможность", то актуализируется первая ее предшествующая пара с меткой „ввод“ в $FG2$ и обрабатывается этот случай. Если для пары с меткой „ввод“ возникает условие „невозможность“, то обрабатывается этот случай; обработка сводится к увеличению содержания актуального $M(I, 3)$ на определенный шаг и изменению $M(I, 2)$ на основе полученного результата. При реализации этих операций $(M(I, 3))$ может выйти из группы в $IF2$, в которой он находился до сих пор. В таком случае актуализируется $M(I-1, K)$ и начинается обработка случая „невозможность“; при этом в $FG2$ актуализируется первая пара с меткой „ввод“, предшествующая актуальной в данный момент паре.

Для получения ответа на запрос $FG2$ просматривается, начиная с определенной позиции. Эта позиция определяется из актуальной вершины в $FG1$.

Алгоритм следит за числом появлений каждой вершины, что позволяет не просматривать полностью $FG1$.

Для ускорения синтеза сложных высказываний осуществляются следующие проверки:

— имеет ли граф-запрос петлю:

— связен ли граф-запрос;

— не превышает ли граф-запрос определенную размерность;

— не пропущена ли вершина в графе-запросе;

— удовлетворены ли условия $((IF1(I, 1)) \geqq (FG1(L, 2))) \& ((IF1(I, 2)) \geqq FG1(L, 3)))$ и $(M(I, 2)) = (M(R, 2))$, где $(M(I, 2))$ — из актуального $M(I, K)$ и $R < I$.

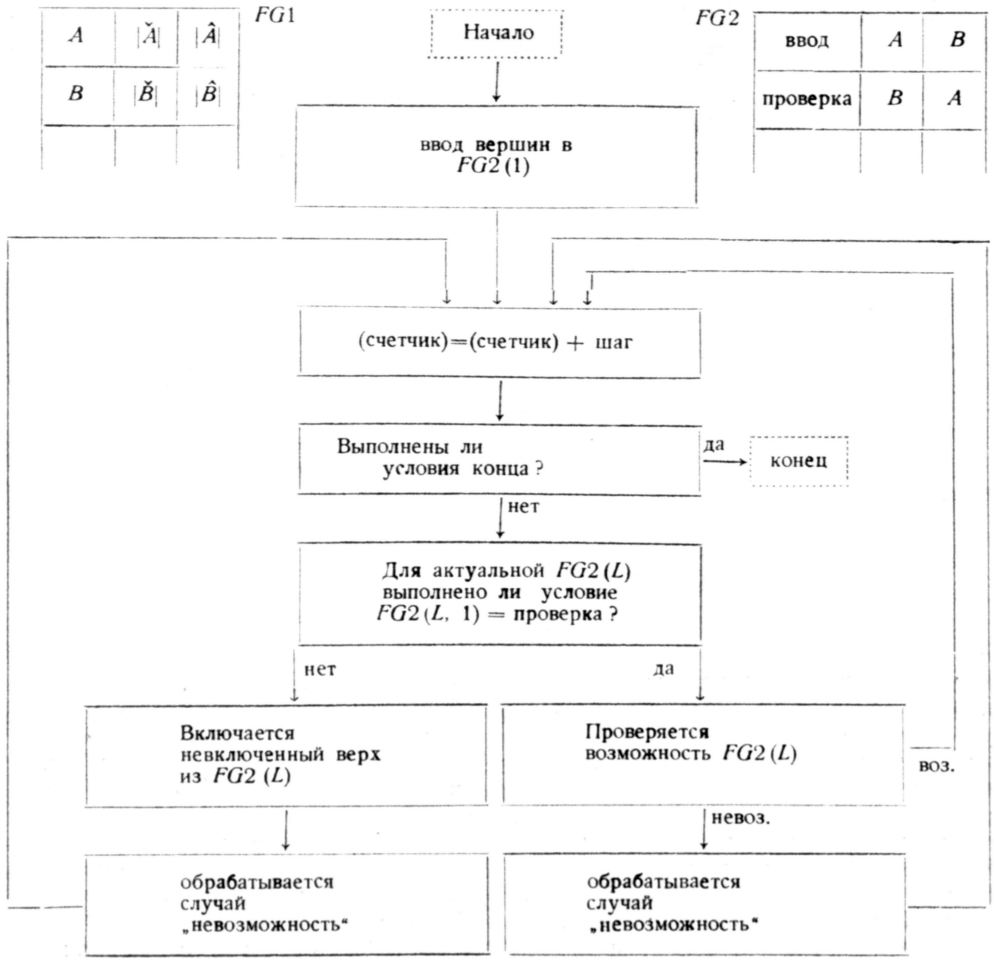


Рис. 3

Внесение некоторых подразумевающихся изменений в предложенном алгоритме позволил бы решать и значительно более сложные задачи, например, синтез сложных высказываний, в которых элементы „вершина" и „дуга" окрашены. Он может быть использован во всех АИС, где информационный массив представим в указанном виде, а также и во всех прикладных задачах поиска изоморфного подграфа. Разработанная программа использована для обработки данных из социологических исследований.

Вычислительный центр
Комитета туризма
София 1000 Болгария

Поступила 13. 9. 1977 г.