

В. Н. КАСЬЯНОВ

ВЫДЕЛЕНИЕ ГАМАКОВ В ОРИЕНТИРОВАННОМ ГРАФЕ

(Представлено академиком Г. И. Марчуком 14 X 1974)

Задача перечисления гамаков ориентированного графа возникает при решении задач автоматического преобразования программ, при решении задачи сегментации, когда программа должна быть разбита на части, так как она целиком не помещается в оперативной памяти и т. д.

Граф программы — это некоторый конечный ориентированный граф $G=(X, \Gamma)$, в котором выделены вершина-вход и вершина-выход, нет дуг, заходящих во вход или исходящих из выхода, и через любую вершину которого проходит путь от входа к выходу. Количество вершин и дуг в графе G будем обозначать соответственно n и m .

Непустое множество $U \subset X$ образует гамак ⁽¹⁾ с начальной вершиной $p \in U$ и конечной $q \notin U$, если $\Gamma^{-1}(U \setminus \{p\}) \subseteq U$, $\Gamma(U) \setminus U = \{q\}$ и $p \notin \Gamma(q)$. Гамак U является линейной компонентой ⁽¹⁾ графа G , если он минимален относительно следующих двух свойств: не существует пути от q к p и вершины p, q принадлежат каждому пути по графу от входа к выходу.

Теорема 1. Существует алгоритм, выделяющий линейные компоненты графа G за st действий.

Теорема 2. Существует алгоритм, выделяющий гамаки графа G за st действий.

Информация о графе задается списками смежности. При этом каждой вершине графа ставится в соответствие два списка: список (непосредственных) предшественников и список (непосредственных) преемников данной вершины. Элементами этих списков являются имена вершин.

Описываемые алгоритмы опираются на подходящие нумерации вершин графа G .

N - и M -нумерации осуществляются при построении пути по графу. При этом M -номера вершин отражают порядок, в котором вершины включаются в путь, а N -номера — порядок, обратный порядку удаления вершин из пути. Процесс построения пути начинается включением входной вершины графа в пустой путь и заканчивается, когда путь снова пуст.

Пусть некоторый путь построен. Для удлинения пути просматривается список преемников его последней вершины до тех пор, пока не будет найдена вершина, которая ранее не включалась в путь. Если такая вершина найдется, путь удлиняется и процесс повторяется; если же нет, то последняя вершина пути удаляется из него и просматривается список преемников предпоследней вершины с того места, на котором этот просмотр был прерван при предыдущем удлинении пути и т. д.

Для хранения пути используется магазин на n двойных слов. В каждый момент выполнения алгоритма в верхушке магазина хранится информация о последней вершине пути: ее имя и имя первого непросмотренного элемента списка ее преемников. Алгоритм требует st действий*.

L -нумерация вначале совпадает с N -нумерацией. Дальнейшее выполнение алгоритма L -нумерации состоит из n одинаковых шагов, в течение каждого из которых производится перенумерация относительно некоторой фиксированной вершины. Выполнение алгоритма перенумерации начи-

* Алгоритм N -нумерации, описанный в ⁽²⁾, требует sn^2 действий.

нается обработкой входной вершины графа, L -номер которой равен 1, и заканчивается перенумерацией относительно вершины, L -номер которой равен n .

После того как зафиксирована некоторая вершина p , строится множество $S(p)$ всех тех вершин, из которых достижима вершина p на графе с удаленными вершинами $\{q: L_q < L_p\}$. Всем вершинам из $S(p)$ в порядке их прежней упорядоченности приписываются новые L -номера $L_p, L_p+1, \dots, L_p+|S(p)|-1$, а всем вершинам $\{q \notin S(p): L_q \geq L_p\}$ также в порядке их прежней упорядоченности — новые L -номера $L_p+|S(p)|, L_p+|S(p)|+1, \dots, n$.

Алгоритм L -нумерации требует sn действий.

Свойства N -нумерации.

Лемма 1. Множество вершин $S \subset X$ образует бикомпоненту (сильно связную компоненту) графа G тогда и только тогда, когда для некоторой вершины $p \notin \bigcup_{N_q < N_p} S(q)$ множество $S(p)$ совпадает с S .

Бикомпоненту S назовем одноходовой, если существует такая вершина $p \in S$, что любой путь от входной вершины графа к вершинам из S содержит p .

Лемма 2. Бикомпонента $S(p)$ одноходова тогда и только тогда, когда не существует вершины из $S(p) \setminus \{p\}$, N -номер некоторого предшественника которой меньше N_p .

Назовем вершину p бивершиной, если $S(p)$ — бикомпонента графа. Пусть $p_1, \dots, p_{k+1}$ — все такие, упорядоченные в порядке возрастания их N -номеров вершины графа G , что $S(p_i)$ — одноходовая бикомпонента и N_{p_i} совпадает с $H(p_i) = \max(1, \{N_q: q \text{ — бивершина и для некоторой бивершины } p \text{ с } N_p < N_{p_i} \cap S(q) \neq \emptyset\})$, и пусть $U_i = \bigcup \{S(q): q \text{ — бивершина и } N_{p_i} \leq N_q < N_{p_{i+1}}\}$, где $i=1, 2, \dots, k$.

Лемма 3. Граф G содержит ровно k линейных компонент: $U_1, \dots, U_k$, где p_i — начальная, а p_{i+1} — конечная вершины линейной компоненты U_i , $i=1, 2, \dots, k$.

Доказательство теоремы 1. Выделение линейных компонент графа осуществляется в два этапа с использованием N -нумерации его вершин.

На первом этапе производится выделение бикомпонент графа (бивершины, образующие одноходовые бикомпоненты, помечаются специальной пометкой). Такое выделение осуществляется при обработке вершин графа в порядке возрастания их N -номеров. На каждом шаге определяется множество $S(p)$ относительно первой непросмотренной вершины p . В процессе выделения $S(p)$ входящие в нее вершины помечаются как просмотренные.

На втором этапе перебираются бивершины в порядке возрастания их N -номеров, вычисляется значение функции H и осуществляется собственное выделение линейных компонент.

Свойства M - и L -нумераций.

Лемма 4. Пусть U — гамак графа G , а p и q — соответственно его начальная и конечная вершины. Тогда

$$U = \{p': L_p \leq L_{p'} < L_p + |U|, \quad M_p \leq M_{p'}\}$$

и либо $M_q < M_p$, либо $L_q = L_p + |U|$.

Пусть

$$U = \{p': L_p \leq L_{p'} < L_p + |U|, \quad M_p \leq M_{p'}\}.$$

Определим

$$\begin{aligned} K(U) &= \{q \in \Gamma(U): M_q < M_p\}, \\ R(U) &= \{q \in \Gamma^{-1}(U \setminus \{p\}): L_q < L_p\}, \\ E(U) &= \max\{L_q: q \in \Gamma(U), M_q > M_p\}, \\ F(U) &= \max\{L_q: q \in \Gamma^{-1}(U \setminus \{p\})\}. \end{aligned}$$

Лемма 5. Множество U образует гамак с начальной вершиной p и конечной вершиной q тогда и только тогда, когда $R(U) = \emptyset$, $F(U) < L_p + |U|$, $p \in \Gamma(q)$ и либо $K(U) = \emptyset$, $L_q = L_p + |U| = E(U)$, либо $K(U) = \{q\}$, $E(U) < L_p + |U|$.

Доказательство теоремы 2. Алгоритм выделения гамаков графа использует M - и L -нумерации его вершин. Выполнение алгоритма состоит из n одинаковых шагов, на каждом из которых осуществляется перечисление всех гамаков для некоторой начальной вершины.

После того как зафиксирована начальная вершина p , помечаются специальной пометкой все предшественники вершины p и по очереди рассматриваются множества последовательно L -занумерованных вершин графа, начинающихся с p и состоящих из одной, двух и т. д. вершин. Это позволяет определять K , R , E и F на всех таких множествах за один просмотр предшественников и преемников вершин графа.

Замечания. Число дуг графа реальной программы сравнимо с числом его вершин (как правило, $m \leq 2n$). С другой стороны, граф без разветвлений, у которого $m = n - 1$, содержит $\frac{n}{2}(n - 1)$ гамаков и $(n - 1)$ линейную компоненту.

При реализации описанных алгоритмов требуется дополнительно не более $6n$ ячеек памяти.

Вычислительный центр
Сибирского отделения Академии наук СССР
Новосибирск

Поступило
14 IX 1974

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

- ¹ И. В. Поттосин, Сборн. Системное и теоретическое программирование, Новосибирск, 1972, стр. 278. ² C. P. Earnest, K. G. Balke, J. Anderson, J. ACM, v. 19, № 1, 23 (1972).