И.Б.Бурдонов, А.С. Косачев.
Построение прямого и обратного остовов автоматами на графе.
Труды Института системного программирования РАН Том 26-6. 2014 г., стр. 57-62.
6 стр.

ПОСТРОЕНИЕ ПРЯМОГО И ОБРАТНОГО ОСТОВОВ АВТОМАТАМИ НА ГРАФЕ

Игорь Бурдонов <igor@ispras.ru>, Александр Косачев kos@ispras.ru

1. Введение

Исследование ориентированных графов является корневой задачей во многих приложениях. В этом докладе мы ориентируемся на такие приложения, как исследование сетей связи, в том числе сети интернета и GRID. Например, в сети интернета WEB-страницу можно понимать как вершину графа, а гиперссылку – как ориентированную дугу. В этом случае исследование графа можно понимать как самоисследование с помощью программ, находящихся в узлах сети, и сообщений, передаваемых между этими программами по дугам графа сети.

Исследование графа, как правило, базируется на классической задаче обхода графа, начиная с выделенной начальной вершиной, которую мы будем называть корнем. Эта задача нетривиальна, если граф ориентирован. Обход ориентированного сильно-связного графа требует времени порядка nm, где n – число вершин графа, а m – число дуг. Такое время обхода достигается многими хорошо известными алгоритмами: обход в глубину, обход в ширину, «жадный» алгоритм и др. [1,2,3].

В 1966 г. М.О. Рабин поставил задачу обхода ориентированного графа конечным автоматом [4]. Автомат на графе аналогичен машине Тьюринга: ячейке ленты соответствует вершина графа, а движение влево или вправо по ленте заменяется переходом по одной из дуг, выходящих из текущей вершины графа. Эту «конструкцию» можно инвертировать: значение в вершине графа понимается как состояние автомата, неподвижно «сидящего» в этой вершине, а по дуге графа, понимаемой как канал передачи, перемещается сообщение, которое автомат в начале дуге посылает автомату в конце дуги.

Сообщения являются одновременно выходными и входными символами автоматов в вершинах. Если размер сообщения и число состояний автоматов в вершинах ограничены, то это случай конечных автоматов. Обычный обход графа – это обход одним сообщением, размер которого не ограничен и является линейной функцией от числа вершин графа. Для конечных автоматов на сегодняшний день наиболее быстрый алгоритм предложен в [5], он имеет оценку $nm+n^2 loglogn$. При повторном обходе автоматы в вершинах находятся не в начальном состоянии, а в состояниях после первого обхода. В этом случае оценка уменьшается до $nm+n^2 l(n)$, где $l(n)$ — число логарифмирований, при котором достигается соотношение $1 \le log(log...(n)...) < 2$ [6].

В данном докладе мы рассматриваем задачу параллельного обхода графа, когда по его дугам может одновременно циркулировать множество сообщений. Оценка времени работы алгоритма зависит от числа сообщений, которые могут одновременно передаваться по дуге. Такое число мы будем называть *ёмкостью дуги* и обозначать k.

Обход графа выполняется за время порядка n/k+D, где D – диаметр графа, т.е. длина максимального пути (маршрута без самопересечений) в графе. В результате обхода будут построены *прямой остов* графа, ориентированный от корня, и *обратный остов*,

ориентированный к корню. Подробное описание алгоритма и доказательства утверждений см. в [7].

2. Алгоритм построения остовов графа

Посылая сообщение, автомат в вершине должен указывать дугу, выходящую из вершины, по которой сообщение посылается. Мы будем считать, что дуги, выходящие из вершины, перенумерованы и автомат указывает номер дуги. Также будем предполагать, что временем срабатывания автомата в вершине можно пренебречь, а время передачи сообщения по дуге ограничено сверху 1 тактом.

Пусть в графе число дуг, выходящих из вершины, не превышает s. Очевидно, $m \le ns$. Мы предлагаем алгоритм построения остовов графа со следующими характеристиками:

- память автомата вершины $O(nD\log s)$,
- размер сообщения $O(D\log s)$,
- ёмкость дуги k,
- время работы алгоритма $O(n/k + D)$.

Выделяют три типа дуг: *прямые дуги* – дуги прямого остова, *хорды* – все остальные дуги, *обратные дуги* – дуги обратного остова. Заметим, что обратная дуга может быть как прямой дугой, так и хордой. *Вектором маршрута* будем называть список номеров дуг маршрута в графе. *Вектор вершины* – это вектор прямого пути до этой вершины. Корень имеет пустой вектор ε. Размер вектора пути или контура $O(D\log s)$. Размер сообщения в описываемом ниже алгоритме равен $O(1)$ векторов, т.е. $O(D\log s)$ бит.

Описание алгоритма мы разобьём на четыре части. Первая часть строит обратный остов, вторая часть – это «стопор», определяющий конец построения обратного остова, третья часть предназначена для разметки типа дуг, выходящих из вершины, четвёртая часть устанавливает в вершинах счётчики входящих обратных дуг.

В первой части используются четыре типа сообщения: ***Старт***, ***Поиск корня***, ***Прямое*** и ***Обратное***. Сообщение ***Старт*** создаётся корнем в начале работы и направляется всем вершинам. Его цель – сообщить каждой вершине её вектор и инициировать рассылку сообщений ***Поиск корня***. Сообщение ***Поиск корня*** рассылается «веером» таким образом, чтобы оно прошло по каждой дуге ровно один раз. Цель сообщений ***Поиск корня*** – пройти некоторый путь от инициатора до корня и сообщить корню вектор этого пути. Корень в ответ посылает сообщение ***Прямое***. Оно направляется по прямому пути до вершины, инициировавшей поиск корня. Его цель – сообщить инициатору *вектор пути* от инициатора до корня. Инициатор создаёт сообщение ***Обратное***, которое размечает обратный путь от инициатора до корня, задаваемый этим *вектором пути*.

Задача второй части – определить конец построения обратного остова. Идея основана на подсчёте в корне числа дуг графа: когда счётчик числа дуг в корне обнулится, обратный остов будет построен. Для каждой дуги $a \to b$ корень сначала получает сообщение от начала дуги a, в котором для дуги $a \to b$ имеется «+1», а заведомо позже – сообщение от конца дуги b, в котором для дуги $a \to b$ имеется «-1». Кроме того, это последнее сообщение от конца дуги b приходит в корень заведомо позже сообщения, посылаемого из b и содержащего «+1» для каждой дуги, выходящей из b. Для этого модифицируется сообщение ***Поиск корня*** и используются два дополнительных сообщения: ***Финиш*** и ***Минус***. В сообщение ***Поиск корня***

добавляется *число дуг*, выходящих из инициатора. Корень при получении этого сообщения прибавляет *число дуг* к своему *счётчику дуг*. Сообщение ***Финиш*** посылается из вершины после получения ею сообщения ***Прямое***, оно разрешает из конца дуги посылать сообщение ***Минус***. Однако ***Минус*** посылается после того, как в вершине появится обратная дуга. Корень при получении этого сообщения вычитает 1 из своего *счётчика дуг*.

Третья часть размечает в каждой вершине выходящие дуги как прямые дуги и хорды. Сначала все выходящие дуги считаются хордами. Затем каждая дуга, по которой посылается сообщение ***Прямое***, отмечается как прямая.

Четвёртая часть устанавливает в вершинах значения *счётчиков входящих обратных дуг*. Для этого используются два сообщения ***Начало подсчёта*** и ***Конец подсчёта***. Сообщение ***Начало подсчёта*** распространяется из корня ко всем вершинам по прямому остову, а сообщение ***Конец подсчёта*** распространяется по обратному остову и в каждой вершине подсчитывается число таких сообщений, созданных в началах входящих обратных дуг.

3. Заключение

В конце работы описанного алгоритма в памяти автомата каждой вершины графа сохраняется следующая информация: тип каждой выходящей дуги и число входящих обратных дуг. Такая разметка графа может использоваться для параллельных вычислений функций от мультимножеств, элементами которых являются значения в вершинах графа.

ЛИТЕРАТУРА:

1. Steven S. Skiena. "The Algorithm Design Manual", Springer-Verlag, New York, 1997.
2. И.Б. Бурдонов, А.С. Косачев, В.В. Кулямин. "Неизбыточные алгоритмы обхода ориентированных графов. Детерминированный случай" // Программирование, 2003 г., №5, с. 59-69.
3. И.Б. Бурдонов, А.С. Косачев, В.В. Кулямин. "Неизбыточные алгоритмы обхода ориентированных графов. Недетерминированный случай" // Программирование, 2004 г., №1, с. 2-17.
4. M.O. Rabin. "Maze Threading Automata. An unpublished lecture presented at MIT and UC", Berkeley, 1967.
5. И.Б. Бурдонов. "Обход неизвестного ориентированного графа конечным роботом" // Программирование, 2004 г., № 4, с. 11-34.
6. И.Б. Бурдонов. "Проблема отката по дереву при обходе неизвестного ориентированного графа конечным роботом" // Программирование, 2004 г., № 6, с. 6-29.
7. И.Б. Бурдонов, А.С. Косачев, В.В. Кулямин. "Параллельные вычисления на графе" // Программирование, 2015 г., №1 (в печати).