**ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ АВТОМАТАМИ ПО ПРЯМОМУ И ОБРАТНОМУ ОСТОВАМ ГРАФА**

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

# 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.

Такое самоисследование графа можно понимать как вычисление некоторой функции от значений в вершинах графа. Поскольку в разных вершинах могут быть записаны одинаковые значения, корректнее говорить о функции от мультимножества. Вычисление инициируется сообщением, приходящим извне в автомат выделенной начальной вершины графа, которую мы будем называть корнем. В ответ автомат корня посылает сообщение, содержащее полученное значение функции.

Как алгоритмы вычисления на графе, так и оценка времени их работы, существенно зависят от того, имеют ли автоматы в начале работы какую-то информацию о графе, или в каждой вершине автомат находится в начальном состоянии и «ничего не знает» о графе, кроме числа дуг, выходящих из вершины. В данной работе мы предполагаем, что в начале работы автоматы находятся в таких состояниях, которые описывают следующую разметку графа. В графе выделены прямой остов, ориентированный от корня, и обратный остов, ориентированный к корню. Дуги, не принадлежащие прямому остову, называются хордами. Обратная дуга может быть как прямой дугой, так и хордой. В памяти автомата вершины для каждой выходящей дуги отмечен её тип: прямая, хорда, прямая и обратная, хорда и обратная. Кроме того, в памяти автомата вершины хранится число входящих обратных дуг. Заметим, что такая разметка графа может быть получена в результате обхода графа, как описано в нашем докладе «Построение прямого и обратного остовов автоматами на графе».

Вычисление функции выполняет *алгоритм пульсации*, основанный на том, что сначала от автомата корня по всему графу распространяются *сообщения-вопросы*, которые должны достигнуть каждой вершины. А затем от каждой вершины «в обратную сторону» к начальной вершине двигаются *сообщения-ответы*. С помощью алгоритма пульсации можно параллельно вычислять любую функцию от мультимножества значений, записанных в памяти автоматов по всем вершинам графа (мы будем говорить «записанных в вершинах»). Время работы алгоритма порядка *D*, где *D* – диаметр графа, т.е. длина максимального пути (маршрута без самопересечений) в графе. Подробное описание алгоритма и доказательства утверждений см. в [].

# Агрегатные функции и агрегатные расширения функций

Алгоритм пульсации, по сути, вычисляет агрегатные функции, для которых значение функции от объединения мультимножеств вычисляется по значениям функции от этих мультимножеств. В этом разделе мы дадим формальное определение агрегатной функции и агрегатного расширения любой функции *f(x)*, которое позволяет представить её в виде *h(g(x))*, где *g* агрегатная функция. Существует и единственное минимальное агрегатное расширение, вычисляющее минимум информации, по которой еще можно восстановить функцию *f*. Теория агрегатных функций, излагаемая в этом разделе, является модификацией теории индуктивных функций в [].

Далее рассматриваются функции на конечных мультимножествах из элементов некоторого базового множества *X*. Множество всех конечных мультимножеств из элементов *X* обозначим через *X*–. Под операцией объединения далее подразумевается операция на мультимножествах, т.е., учитывающая кратности элементов.

*Агрегатная функция* *g:X–→A* – это такая функция, что

*∃e:A×A→A ∀a,b∈X– g(a∪b) = e(g(a),g(b))*.

*Агрегатным расширением* функции *f:X–→A* назовём агрегатную функцию *g:X–→B*, такую, что *∃h : B→A ∀a∈X– f(a) = h(g(a))*.

Агрегатное расширение *g* функции *f* представляет собой агрегатную функцию, по которой можно вычислить функцию *f*. При этом возможны такие расширения, которые на практике не помогают в этом, например, можно взять в качестве *g* тождественную функцию на *X–*, а в качестве *h* – саму функцию *f*. Чтобы избежать такого, используется минимальное агрегатное расширение; интуитивно, это агрегатная функция, вычисляющая минимум информации, по которой еще можно восстановить *f*.

Агрегатное расширение *g : X–→B* функции *f : X–→A* назовём *минимальным*, если *g(X–) = B* и *∀g` : X–→C*, являющегося агрегатным расширением *f*, имеет место *∃i : C→B g = ig`*.

Минимальное агрегатное расширение функции *f : X–→A* существует и единственно с точностью до взаимно однозначных отображений.

# Алгоритм пульсации

Алгоритм пульсации предназначен для вычисления значения заданной функции *f* от мультимножества *x∈X–* значений, записанных в вершинах графа. Мы будем предполагать, что в каждой вершине *i* записано значение *x(i)* с единичной кратностью. Алгоритм пульсации использует описанную во Введении разметку графа.

Алгоритм пульсации использует два типа сообщений: ***Вопрос*** и ***Ответ***. Сначала в автомат корня поступает извне сообщение ***Вопрос***, содержащее указание на три функции: *h*, *e* и *g*. Это сообщение распространяется по прямому остову, чтобы все вершины запомнили параметры *e* и *g*. Вершина обратного остова вычисляет значение функции *g* от значений во всех вершинах поддерева обратного остова с корнем в данной вершине и посылает его как параметр сообщения ***Ответ*** по обратной дуге. Корень, вычислив *g(x)*, посылает вовне сообщение ***Ответ*** с параметром *f(a)=h(g(x))*.

Время работы алгоритма пульсации не превышает ***O****(D)*.

**ЛИТЕРАТУРА:**

1. И.Б. Бурдонов, А.С. Косачев, В.В. Кулямин. "Параллельные вычисления на графе" // Программирование, 2015 г., №1 (в печати).
2. Кушнеренко А.Г., Лебедев Г.В. "Программирование для математиков", Наука, Главная редакция физико-математической литературы, Москва, 1988.