Записки научных
семинаров ПОМИ
Том 407, 2012 г.

Н. К. Косовский

ПОЛИНОМИАЛЬНО ОГРАНИЧЕННЫЙ СВЕРХУ ОБЪЁМ ИЗМЕНЕНИЙ ПРОГРАММ НА RAM+BOOL ДЛЯ ДОКАЗАТЕЛЬСТВА ПРИНАДЛЕЖНОСТИ FP

Эта статья посвящается светлой памяти Николая Александровича Шанина, моего учителя, который почти всегда в публичных выступлениях подчёркивал важность исследования эффективности вычислений, под которой за последние 30 лет многие математики, работающие в области математической теории информатики, стали понимать вычислимость за полиномиальное число шагов на машинах Тьюринга (класс **FP**). Этим определяется мотивировка полезности применения теорем настоящей статьи для доказательства полиномиальной эффективности различных алгоритмов, реализованных на компьютере.

RAM+BOOL (т.е. RAM [1] с дополнительными поразрядными логическими операциями) кажется достаточно подходящим математическим понятием алгоритма для того, чтобы более адекватно (относительно операций компьютеров) отразить число шагов вычисления, чем понятие машины Тьюринга. Это связано с тем, что в RAM имеется, например, косвенная адресация, широко используемая при обработке массивов в компьютере. Однако за n шагов на RAM+BOOL можно вычислить, например, 2^{2^n}, что гораздо менее адекватно числу шагов вычисления этой функции на компьютере.

Более того, из работы [2] следует, что на RAM+BOOL за полиномиальное число шагов (при равномерном весе команд [1]) можно решить задачу ВЫПОЛНИМОСТЬ, которая является NP-полной задачей [4].

Ниже для обеспечения принадлежности классу **FP** (т.е. классу алгоритмов, вычислимых за полиномиальное время на машине Тьюринга) предлагается использовать введенное в [3] понятие объёма изменений, представляющее собой произведение числа шагов вычисления на длину используемой памяти.

Под использованной памятью (при логарифмическом весе) RAM+BOOL будем понимать максимум по всем шагам вычисления

Ключевые слова: RAM, машины Тьюринга, машины Тьюринга с оракулами-функциями, класс FP.

программы на RAM+BOOL сумм длин всех значений содержимых всех использованных ячеек.

Предложенная характеризация алгоритма посредством объёма изменений представляет собой ресурсно-объединённую характеризацию, полиномиальная верхняя граница которой, во-первых, удобна для её вычисления и, во-вторых, как будет доказано ниже, позволяет доказать полиномиальную верхнюю границу числа шагов реализации того же алгоритма на машине Тьюринга, измеряемая относительно длины записи исходных данных, записанных в двоичной системе счисления.

Программирование на компьютере, как правило, использует процедуры-функции. Исключением является чаще всего микропрограммирование контроллеров.

Простейшим математическим аналогом использования процедур-функций является машина Тьюринга с оракулом-функцией (см., например, [5]). В этом случае конечное число процедур-функций может быть заменено одной новой процедурой-функцией путем введения дополнительного аргумента — номера первоначальной процедуры-функции.

При подсчёте числа шагов машины Тьюринга с оракулом-функцией применение оракула считается за один шаг, за который происходит считывание с входной оракульной ленты, очищение выходной оракульной ленты, затем запись результата применения оракула-функции на выходную оракульную ленту и очищение входной оракульной ленты. Удобно считать оракульные ленты односторонними. Память оракульной машины Тьюринга подсчитывается здесь как максимальная длина используемых лент, включая оракульные.

Удобно считать, что в языке RAM+BOOL поразрядные операции всегда возвращают положительное число (представленное в двоичной системе счисления). При этом у отрицательных аргументов отбрасывается знак, а число 0 преобразуется в 10. По существу, к битовым строкам, к которым применяются поразрядные логические операции, в качестве самого старшего разряда (слева) приписана 1. Перед применением двухместной логической операции к аргументу с меньшей длиной непосредственно после начальной 1 вставляется необходимое количество нулей для уравнивания длин аргументов. Достаточно использовать общепринятые поразрядные логические операции: конъюнкцию, дизъюнкцию и отрицание.

Лемма. *Каждая из операций RAM+BOOL, включая косвенную адресацию, используемую в рамках полиномиально ограниченного сверху от длины записи исходных данных массива чисел, принадлежит классу* ***FP****.*

Доказательство. Традиционные алгоритмы выполнения операций RAM+BOOL могут быть реализованы на машине Тьюринга в рамках класса **FP**. Это же верно и для косвенной адресации, применяемой в пределах полиномиально ограниченного массива чисел отдельно для каждой степени полинома. При этом массив реализуется в виде списка пар (номер элемента массива, содержимое элемента массива с этим номером), расположенных в порядке возрастания номеров элементов. □

Теорема 1. *Каждая программа, написанная на языке RAM+BOOL и работающая за полиномиальное число шагов (с равномерным весом команд) на полиномиальной ленте (при логарифмическом весе), для реализации алгоритма f, может быть реализована на машине Тьюринга с оракулом-функцией, вычисляющей по номеру операции RAM+BOOL и аргументам, включая содержимое массива, результат каждой из операций , включая косвенную адресацию, так, что число шагов и память такой машины Тьюринга ограничена сверху полиномом от длины записи исходных данных. При этом её оракул-функция принадлежит* ***FP****.*

Доказательство. Все операции RAM+BOOL могут быть объединены в одну оракул-функцию, преобразующую регистр и массив целых чисел, представленный в виде списка пар (номер элемента массива, содержимое элемента массива с этим номером), расположенный в порядке возрастания номеров ячеек (адресов ячеек).

По лемме и условию теоремы такой оракул-функция принадлежит **FP**. Следовательно, алгоритм f вычислим на машине Тьюринга с оракулом-функцией, принадлежащим **FP**, за число шагов и памятью, ограниченные сверху полиномом от длины записи исходных данных. □

Следствие. *Класс всех программ, написанных для RAM+BOOL, объём изменений которых ограничен сверху полиномом от длины записи исходных данных, совпадает с классом* ***FP****.*

Замечание. В формулировке следствия объём изменений нельзя заменить ни на число шагов (при равномерном весе команд), ни, вообще говоря, на объём памяти (при логарифмическом весе, а тем более при равномерном весе). В первом случае опровергающий пример приведен в начале статьи (вычисление 2^{2^n}). Во втором случае удалось бы вычислить каждый предикат из **P-SPACE** в рамках **FP** и, следовательно, **P**. Тем самым, в частности, удалось бы доказать, что **P=NP**, решив тем самым одну из фундаментальных проблем теории сложности алгоритмов.

Доказательство следствия. Достаточно заметить, что объём изменений превосходит как число шагов, так и память, необходимую при вычислении, поскольку является их произведением. Поэтому к доказанной теореме достаточно применить теорему из [3]. Приведём формулировку этой теоремы.

Теорема. *Пусть оракул-функция принадлежит классу **FP**. Тогда классу **FP** (классу **P**) принадлежит функция, вычислимая на машине Тьюринга с оракулом-функцией, объем изменений которой не превосходит полинома от длины записи исходных данных (соответственно длина записи результата вычисления которой ограничена квазилинейной функцией от длины записи аргумента, и соответственно значение которой принадлежит множеству* $\{0,1\}$*).*

Кроме этого необходимо проверить, что работа машины Тьюринга, решающей задачу из класса **FP**, может быть промоделирована на RAM+BOOL с полиномиальной верхней границей изменения объёма.

Массив можно использовать как одностороннюю ленту машины Тьюринга, 0 как бланковый символ, конечное число положительных целых чисел– как буквы внешнего алфавита машины Тьюринга, конечное число отрицательных чисел – как внутренний алфавит (алфавит состояний машины Тьюринга, совмещенный с головкой машины Тьюринга, указывающей на ячейку с номером, увеличенным на единицу по сравнению с чем номером той ячейки, в которой записано отрицательное число).

В этом случае реализация на RAM+BOOL как очередного, так и заключительного шага машины Тьюринга не вызывает затруднений. □

Пусть **FP-S-CH**(RAM+BOOL) — класс алгоритмов, вычислимых посредством RAM+BOOL за полиномиальный объём изменений (size of change).

Для более простого способа получения верхней полиномиальной оценки памяти (с логарифмическим весом) можно использовать произведение максимального номера (адреса) используемой ячейки на максимальную длину содержимого среди всех ячеек. В этом случае класс **FP-S-CH**(RAM+BOOL) превращается в класс **FP-S-CH**$^+$(RAM+BOOL), для которого доказательства аналогов теоремы 1 и его следствия из настоящей статьи могут быть получены незначительной модификацией приведённых здесь доказательств.

Таким образом, получаем равенство.

Теорема 2. *FP-S-CH*$^+$*(RAM+BOOL) =* ***FP-S-CH****(RAM+BOOL).*

На основе результатов настоящей статьи для доказательства принадлежности классу **FP** алгоритма, реализованного на компьютере, достаточно доказать его принадлежность классу **FP-S-CH**$^+$(RAM+BOOL), что значительно легче, чем дополнительная реализация этого алгоритма на машине Тьюринга.

Литература

1. А. Ахо, Дж. Хопкрофт, Дж. Ульман, Построение и анализ вычислительных алгоритмов. М., Мир, 1979.
2. Н. К. Косовский, Т. М. Косовская, *Полиномиальность и NP-трудность задачи вычисления знака постоянного терма.* — Математические вопросы кибернетики, вып.16 (2007), 125–128.
3. Н. К. Косовский, *Условия полиномиальности алгоритмов, реализуемых на машинах Тьюринга с оракулами-функциями.* — Материалы XVII Международной школы-семинара “Синтез и сложность управляющих систем” им. акад. О.Б.Лупанова (Новосибирск, 27 октября – 1 ноября 2008 г.). Новосибирск: Изд-во Института математики, 2008, 70–74.
4. С. А. Кук, *Сложность процедур вывода теорем.* — Киб. сб., нов. сер., вып. 12 (1975), 5–15.
5. D.-Z. Du, K.-I. Ko, Theory of Computational Complexity. A Wiley-Interscience Publication. John Wiley & Sons, Inc. 2000.

Kosovskiy N. K. Polynomial upper bounds of RAM+BOOL program size of changes for the proof of belonging to FP.

RAM+BOOL (i.e. a random access machine with the additional bitwise logical operations) seams to be suitable enough mathematical notion of

an algorithm for a more adequate (according to the computer operations) simulation of the number of computational steps than the notion of Turing machine. For example, the definition of RAM+BOOL contains indirect address widely used while array processing in computer. The RAM+BOOL used memory size (with logarithmical weight) is maximum over steps of a RAM+BOOL program run of the summs of the lengths of all values of all cells.

The introduced by the author notion of the size of changes (which is a production of the number of steps and the used memory size) is offered to use in order to guarantee the belonging of RAM+BOOL computations to the class **FP** (i.e. to the class of all algorithms computable by a Turing machine in a polynomial time).

Some statements concerning complexity characteristics of RAM+BOOL computations are proved in the paper. The most important of them is the following corollary.

Corollary. *The class of all RAM+BOOL programs, the size of chanches of which has a polynomial under the input data length upper bound, coincides with the class* ***FP****.*

С.-Петербургский
государственный университет,
Университетский пр. 28, Петродворец,
198504 Санкт-Петербург, Россия
E-mail: kosov@nk1022.spb.edu

Поступило 16 апреля 2012 г.