



## ТЕЗИС ЧЁРЧА ДЛЯ ПОЛИНОМИАЛЬНЫХ ПО ВРЕМЕНИ РЕКУРСИВНЫХ АЛГОРИТМОВ НАД СЛОВАМИ И ИХ ДЛИНАМИ

### Аннотация

Статья содержит два основных раздела. Первый из них (п. 2) отвечает на вопрос о том, как можно расширить понятие нормального алгоритма Маркова, включив, в частности, полиномиальные функции над длинами используемых слов и сохранив при этом объём вычислений в пределах полиномиального числа шагов. Ответом на этот вопрос является последовательность математических понятий алгоритма со всё более мощными вычислительными средствами, но тем не менее совпадающими с классом алгоритмов, полиномиальных по времени (с классом **FP**) при реализации их на машинах Тьюринга.

Второй из основных разделов (п.3) отвечает на вопрос, каким образом можно, всё более расширяя понятие нормального алгоритма Маркова, вложить их в каждый из подклассов **FP**, алгоритмы которого используют промежуточную память, длина которой ограничена полиномом  $k$ -ой степени. Результаты второго раздела могут рассматриваться также и как подтверждение естественной формулировки тезиса Чёрча для последнего из классов алгоритмов, использованных в формулировках теорем этого раздела.

**Ключевые слова:** полиномиальные верхние оценки числа шагов алгоритма, класс **FP**, полиномиальные верхние оценки памяти, подклассы **FP-SPACE**, нормальные алгоритмы Маркова, правила Поста.

### ВВЕДЕНИЕ

Получение доказательства полиномиальных верхних оценок числа шагов рекурсивных алгоритмов сопряжено со значительными сложностями, связанными, в частности, с особенностями основных математических понятий алгоритмов, не предназначенными для таких доказательств.

Язык программирования рефал-5 [7] можно рассматривать как точное, прежде всего рекурсивное, математическое понятие алгоритма для вычислений над словами и их длинами. Однако описание этого языка слишком длинно с точки зрения

математики, поэтому некоторые полезные для эффективного программирования средства можно исключить, существенно упростив тем самым формулировку математического понятия алгоритма и предназначив его для описания полиномиальных по числу шагов математических рекурсивных преобразований и вычислений над словами и их длинами. Предлагаемые понятия включают в себя нормальные алгоритмы Маркова (см., например, [6]) и алгоритмы Маркова-Поста [5] с ограничениями, описываемыми здесь.

Полиномиальный (по времени) тезис Чёрча для предлагаемых в разделе 2 понятий алгоритма может быть сформулирован следующим образом.

*Класс всех интуитивно полиномиально ограниченных по числу шагов алгоритмов над словами и их длинами совпадает с классом каждого из предлагаемых понятий алгоритма, которые всегда заканчивают свою работу не более чем за полиномиальное число шагов.*

Под использованной памятью алгоритма при вычислении его аргумента понимаем максимальную длину записи среди элементов последовательности всех промежуточно вычисляемых постоянных (то есть не содержащих переменных) выражений.

Будем говорить, что используемая память алгоритма полиномиальна  $k$ -ой степени, если для любых исходных данных длина всех её промежуточных значений при вычислениях (включая аргумент и результат) ограничена сверху полиномом  $k$ -ой степени от длины записи этих исходных данных.

Под шагом алгоритма понимаем успешное выполнение правила этого алгоритма.

Полиномиальный по времени и квазилинейный по памяти тезис Чёрча для самого последнего из предлагаемых понятий алгоритма может быть сформулирован следующим образом.

*Класс всех алгоритмов, интуитивно ограниченных полиномиально по числу шагов и квазилинейно по длине памяти, совпадает с классом используемых в п. 5) теоремы 7 понятий алгоритма, всегда работающих в пределах полиномиального числа шагов и квазилинейной памяти.*

Можно сравнить предложенную формулировку тезиса Чёрча с иной формулировкой его для рефал-5 из [3, 5] и для машин Тьюринга из [9].

## 1. ПРЕДВАРИТЕЛЬНЫЕ ОПИСАНИЯ

Для удобства записи в виде строки символов совокупность всех правил (подстановок) нормального алгоритма Маркова будем заключать в фигурные скобки, отделяя сами правила друг от друга точкой с запятой.

Применение к перерабатываемому слову однопосылочного правила Поста с переменными для слов ко всему перера-

батываемому слову (а не к его части, как в нормальных алгоритмах) детерминируем следующим образом.

- Все переменные для слов упорядочиваются по первым их вхождением в посылку правила.

- В этом порядке переменным для слов в цикле присваиваются значения, начиная с пустого слова. Первая переменная соответствует самому внешнему циклу. Последняя переменная соответствует последнему по вложенности циклу.

- На каждом шаге любого цикла после неудачного сравнения с перерабатываемым словом длина значения переменной цикла увеличивается на единицу. «Успешное» сравнение вызывает замену перерабатываемого слова на заключение правила Поста с подстановкой «успешных» значений вместо каждого вхождения всякой переменной для слов.

- В случае, когда длина значения переменной для слов больше длины всего перерабатываемого слова, её значение, а также значения всех последующих по упорядочиванию переменных заменяются на пустое слово. Если эта переменная является первой в правиле (то есть она соответствует самому внешнему циклу), то правило не применимо, в противном случае длина записи значения (предыдущей по упорядочиванию) переменной увеличивается на единицу.

Удобно разрешить использование переменных для букв (символов), используя префикс  $S$  в записи имён этих переменных. Префикс  $e$  с индексом используется в записи переменных для слов. Используемые в правилах слова удобно заключать в апострофы.

Пусть  $FP$  – класс алгоритмов, вычисляемых на машинах Тьюринга [1, 2] за полиномиальное число шагов.

## 2. СЕРИЯ РАСШИРЯЮЩИХСЯ ПОНЯТИЙ АЛГОРИТМОВ И ТЕОРЕМ ДЛЯ НИХ С ПОЛИНОМИАЛЬНЫМИ (ПО ВРЕМЕНИ) ОГРАНИЧЕНИЯМИ СВЕРХУ

Следующее определение является основополагающим для первых двух теорем.

**Определение.** Алгоритмом Маркова-Поста без увеличения числа вхождений каждой переменной для слов назовём нормальный алгоритм только с односсылочными правилами Поста, если в каждом его правиле, не являющемся заключительным, число вхождений каждой переменной для слов в заключение правила не превосходит числа её вхождений в посылку этого правила.

**Теорема 1.** Класс  $FP$  совпадает с классом всех алгоритмов Маркова-Поста без увеличения числа вхождений каждой переменной для слов и работающих в пределах полиномиального числа шагов.

Теорема 1 является слегка изменённой формулировкой следствия 3 из [5].

Разрешим именовать алгоритмы Маркова-Поста, ставя не содержащее пробел имя перед записью тела алгоритма, заключённого в фигурные скобки, и записывать их применение в виде  $\langle$  Имя Аргумент  $\rangle$ . Такое применение (включая рекурсивное обращение) разрешим использовать (в том числе и вложено) в правой части правила. В этом случае отпадает необходимость использования точки для выделения каждого заключительного правила, если правая часть каждого не заключительного правила заменена рекурсивным обращением к ней.

Так определённые алгоритмы назовём рекурсивными алгоритмами Маркова-Поста.

Для этого понятия алгоритма (и последующих, о которых пойдет речь в этом разделе) можно говорить о таких определяемых алгоритмах, которые определяются с помощью вспомогательных алгоритмов, принадлежащих  $FP$ , и верхняя граница длины записи каждого результата которых не превосходит длины записи соответствующего аргумента, сложенного с длиной записи значения линейной функции (одной и той же для всех результатов) от длины записи аргумента.

Класс всех функций из  $FP$ , для которых выполняются указанные ограничения (как, впрочем, и сам класс  $FP$ ), замкнут относительно операции подстановки.

При этом при подсчёте числа шагов каждого алгоритма число шагов, выпол-

ненное любым вспомогательным алгоритмом, считается равным единице.

Вычисление каждого постоянного выражения всегда начинается с преобразования самого левого подтерма, начинающегося с левой угловой скобки и кончающегося правой угловой скобкой, а также не содержащего внутри себя ни одного другого вхождения угловой скобки (ни левой, ни правой).

**Теорема 2.** Класс  $FP$  совпадает с классом всех алгоритмов, написанных на языке рекурсивных алгоритмов Маркова-Поста без увеличения числа вхождений каждой переменной для слов и работающих в пределах полиномиального числа шагов.

Подробное доказательство этой теоремы имеет большую длину, ибо, по существу, требует моделирования полиномиальной рекурсивной функции полиномиальной итерацией, что может быть осуществлено с помощью задержанных вычислений А.П. Ершова. Для этого каждый раз вычисляется самый левый атомарный терм, то есть терм, аргументы которого не содержат никаких функторов.

Теорема 2 является усилением следствия 5 из [5].

Условие применения правила может быть типа «словарный терм:словарный терм» или типа « $\langle$  Функтор Аргумент  $\rangle$ :словарный терм». Здесь двоеточие читается как «вида». Аргумент может содержать переменные из словарного термина, предшествующего этому условию, но находящемуся в том же правиле. Словарный терм после двоеточия (так же как и посылку правила Поста) удобно называть образцом (или образцом для сравнения).

Для рекурсивных алгоритмов Маркова-Поста с охраняемыми правилами разрешим использование последовательности условий применения каждого правила, которые разделяются запятой. Эта последовательность может находиться после посылки правила с последующей запятой и заканчиваться непосредственно перед стрелкой. Примером последовательности из образца с двумя условиями является  $e_1'A'e_2, e_1:e_3'Ve_4, e_3e_4:e_2$ .

Считаем, что в условиях применения правил отсутствуют рекурсивные вызовы определяемой функции.

Теперь организация циклов по длине слов, подставляемых вместо переменных в качестве их значения, распространяется и на выполнение всех условий, которые проверяются последовательно слева направо. При неудаче выполнения хотя бы одного условия длина записи значения соответствующей переменной цикла увеличивается на единицу, если это возможно (см. начало этого раздела). По существу, такое использование последовательности условий является упрощённым и эффективным вариантом отката (бэктрекинга) языка пролог.

Следующее определение является основополагающим для теоремы 3.

**Определение.** Назовём рекурсивный алгоритм Маркова-Поста с охраняемыми правилами рекурсивным алгоритмом Маркова-Поста с охраняемыми правилами без увеличения числа вхождений переменных для слов, если для каждой переменной для слов из самого левого образца или из охраняемого условия, где имеется ненулевое число вхождений этой переменной, это число не превосходит числа её вхождений в это правило после этого образца.

**Теорема 3.** Класс *FP* совпадает с классом всех алгоритмов, работающих в пределах полиномиального числа шагов и написанных на языке рекурсивных алгоритмов Маркова-Поста с охраняемыми правилами без увеличения числа вхождений переменных для слов.

Разрешим конец каждого правила (замены), начиная с образца последнего охраняемого условия, заменять на запись рекурсивного алгоритма Маркова-Поста без имени. Разрешим также многократное и вложенное применение этого изменения. Полученный класс алгоритмов назовём классом рекурсивных алгоритмов Маркова-Поста с древовидными образцами.

Следующее определение является основополагающим для теорем 4 и 5.

**Определение.** Назовём рекурсивный алгоритм Маркова-Поста с древовидными правилами рекурсивным алгоритмом Маркова-

Поста с древовидными правилами без увеличения числа вхождений переменных для слов, если для каждой переменной для слов из самого левого образца или из охраняемого условия (в образце из ветви, которая начинается с посылки правила Поста и кончается его заключением), где имеется ненулевое число вхождений этой переменной, это число не превосходит числа её вхождений в эту ветвь правила после этого образца.

**Теорема 4.** Класс *FP* совпадает с классом всех алгоритмов, работающих в пределах полиномиального числа шагов и написанных на языке рекурсивных алгоритмов Маркова-Поста с древовидными правилами без увеличения числа вхождений переменных для слов.

Разрешим использование функций, вычисляющих длины записи значений целочисленных полиномов от длин словарных термов. При этом положительное целое число представляется в двоичной системе счисления последовательностью двоичных цифр, начинающейся с единицы. Ноль представляется цифрой ноль. Отрицательное число начинается знаком минус, за которым следует непустая последовательность двоичных цифр, начинающаяся с единицы.

Напомним, что слова (в том числе и расположенные между числами) заключаются в апострофы, если они находятся в записи алгоритма.

Если  $R$  – слово, то  $\langle \| R \| \rangle$  – команда для вычисления его длины, записанной в системе счисления по основанию 2, хотя можно было бы использовать и систему с большим основанием.

Для вычисления значения полиномиального целочисленного выражения от длин аргументов оно заключается в угловые скобки вида  $\langle \rangle$ .

Пусть имя функции, не содержащее пробелов, записывается с прописной буквы.

Функция *Number* переводит слово, соответствующее числу, в число. Например,  $\langle \text{Number } '10' \rangle = -10$ .

Функция *Symbol* является обратной к *Number*. Например,  $\langle \text{Symbol } 10 \rangle = '10'$ .

Удобно двоичную цифру считать символом при использовании переменных

как для слов, так и для букв (то есть для символов). Таким образом, длина записи неотрицательного двоичного числа совпадает с суммарным числом цифр в нём.

Сформулированные к этому моменту средства математической алгоритмизации по существу представляют собой структурированные средства рекурсивного программирования над словами и их длинами (на основе алгоритмов Маркова-Поста и линейных выражений над произвольными целыми числами). Это позволяет описывать многие математические алгоритмы над словами и их длинами, в том числе и доказывать их вычислимость на машинах Тьюринга в пределах полиномиального относительно длины записи исходных данных числа шагов, что и утверждает следующая теорема.

**Теорема 5.** *Класс  $FP$  совпадает с классом всех алгоритмов, работающих в пределах полиномиального числа шагов и написанных рекурсивными алгоритмами Маркова-Поста с древовидными правилами и со встроенными функциями  $Number$  и  $Symbol$ , функциями вычисления как длины, так и полиномиальных выражений над длинами словарных термов без увеличения числа вхождений каждой переменной для слов.*

Полное доказательство всех теорем оказывается достаточно длинным, поскольку оно основано на полиномиальной реализации всех изложенных средств математической алгоритмизации посредством нормальных алгоритмов Маркова-Поста. Наконец, последние алгоритмы полиномиально реализуются нормальными алгоритмами Маркова, которые, в свою очередь, реализуются в пределах полиномиального числа шагов машины Тьюринга.

### 3. ТЕОРЕМЫ ДЛЯ ДОПОЛНИТЕЛЬНЫХ ВАРИАНТОВ ПРЕДЛАГАЕМЫХ ПОНЯТИЙ АЛГОРИТМОВ В ПРЕДЕЛАХ ПОЛИНОМИАЛЬНО ОГРАНИЧЕННЫХ ЧИСЛА ШАГОВ И ДЛИНЫ ЗАПИСИ ИСПОЛЪЗУЕМОЙ ПАМЯТИ

Важно различать функцию и реализующий её алгоритм. С точки зрения по-

лучения одновременных (совмещённых) оценок сложности, эти понятия различны. Например функция может принадлежать классу  $FP$  и классу  $FLIN-SPACE$ , то есть классу  $FP \cap FLIN-SPACE$ , при этом один из реализующих её алгоритмов принадлежит классу  $FP$ , а другой – классу  $FLIN-SPACE$ . Здесь и далее  $FLIN-SPACE$  – это класс алгоритмов, вычисляемых на машине Тьюринга в пределах линейной памяти.

Функция принадлежит классу  $FP \parallel FLIN-SPACE$ , если имеется алгоритм, реализующий эту функцию, который одновременно принадлежит как классу  $FP$ , так и классу  $FLIN-SPACE$ .

Обозначение  $\parallel$  удобно читать как «и одновременно», или как «совмещённый с», или «соединённый с». Очевидно, что  $FP \parallel FLIN-SPACE \subseteq FP \cap FLIN-SPACE$ .

Интересен вопрос о включении при каждом положительном целом  $k$  с помощью вводимых ниже алгоритмов в классы  $FP \parallel n^k-SPACE$ , то есть классы тех алгоритмов из  $FP$ , которые в рекурсивных вызовах для записи промежуточных результатов вычисления используют только память, длина которой не превосходит некоторого полинома  $k$ -ой степени от длины записи исходных данных.

Для всякого  $k \geq 1$  в класс  $FP \parallel n^k-SPACE$  удаётся вложить некоторые классы алгоритмов со всё более разнообразными средствами математической алгоритмизации.

Под алгоритмами Маркова-Поста по-прежнему понимаются нормальные алгоритмы, в которых каждое правило подстановки (в том числе и заключительные правила) заменены на однопосылочные детерминированные правила Поста.

В этом разделе удаётся ограничение на отсутствие увеличения числа вхождений переменных для слов заменить на отсутствие вхождений переменных для слов в правой части правила Поста, которые отсутствуют в левой части этого правила.

При каждом положительном целом  $k$  в рекурсивных алгоритмах Маркова-Поста без использования дополнительной (степени большей  $k$  при  $k \geq 1$ ) полиномиальной памяти в записи однопосылочных

детерминированных правил Поста каждое заключение может использовать выражения, содержащие как не более одного вхождения вызова (рекурсивного) определяемой функции, так и любое число вызовов вспомогательных функций только из  $\mathbf{FP}\|LIN\text{-SPACE}$ . Дополнительные ограничения на значения вспомогательных функций в этом разделе отсутствуют.

Напомним, что в определении понятия рекурсивного алгоритма Маркова-Поста можно обойтись без точки, отмечающей заключительные правила (так как они не содержат рекурсивного вызова, а остальные правила должны его содержать).

При каждом  $k \geq 1$  имеет место включение  $\mathbf{FP}\|n^k\text{-SPACE} \subsetneq \mathbf{FP}\|n^{k+1}\text{-SPACE}$ . Перемножение  $k + 1$  полиномов является примером функции, принадлежащей классу  $\mathbf{FP}\|n^{k+1}\text{-SPACE}$ , но не принадлежащей классу  $\mathbf{FP}\|n^k\text{-SPACE}$ .

Во всех определяемых в статье понятиях рекурсивных алгоритмов используется традиционный для математики и для языка рефал порядок вычисления термов, содержащих имена используемых алгоритмов: начиная с самого левого терма, аргумент которого не содержит никаких имён алгоритмов.

Напомним, что под атомарным термом понимаем терм, начинающийся сразу после левой угловой скобки с единственного вхождения имени функции, входящего в него.

Под длиной записи постоянного терма понимаем суммарное количество символов в его записи.

Пусть длина записи промежуточных вычислений полиномиального целочисленного выражения от длин записи аргументов, являющихся словами, определяется как максимум длин всех его естественных изменений, каждый раз начиная с самых левых среди самых внутренних скобок (если они есть), а внутри них с самых левых среди самых приоритетных операций в традиционном порядке этих приоритетов: возведение в константную степень, умножение и, наконец, аддитивные операции одинакового приоритета сложение и умножение.

Длина записи промежуточных вычислений функций *Number* и *Symbol* равна длине записи их аргументов.

**Определение.** Под длиной записи промежуточных вычислений определяемой функции понимаем максимум среди длины записи аргумента, длины записи результата и длин записей всех постоянных термов (точнее, правых частей правил этой определяемой функции, в которые вместо переменных подставлены соответствующие значения), изменяемых в процессе вычисления путём последовательной замены в них (по соответствующим правилам) каждый раз самого левого вхождения атомарного терма. Если это атомарный терм, начинающийся с имени вспомогательной (а не определяемой) функции, то к аргументам максимума добавляется и длина записи промежуточных вычислений этого атомарного терма, сложенная с длиной записи оставшейся (неизменяемой в этот момент) части всего постоянного терма.

По-прежнему считаем, что в записи условий применения правил отсутствуют рекурсивные вызовы определяемой функции.

**Теорема 6.** При всяком положительном целом  $k$  имеет место последовательность включений следующих классов алгоритмов:

1) класс алгоритмов Маркова-Поста, вычисляемых не более чем за полиномиальное степени  $k$  число шагов и одновременно в пределах длины записи промежуточных вычислений, не превосходящей некоторого полинома  $k$ -ой степени от длины записи аргумента;

2) класс таких же, но рекурсивных алгоритмов, использующих вспомогательные функции только из  $\mathbf{FP}\|n\text{-SPACE}$  с теми же ограничениями и, кроме того, допускающих вычисление полиномов над длинами слов;

3) класс таких же алгоритмов с теми же ограничениями, но, возможно, ещё и с охраняемыми правилами;

4) класс таких же алгоритмов с теми же ограничениями, но, возможно, ещё и с древовидными правилами;

5) класс алгоритмов Маркова-Поста, вычисляемых не более чем за полиномиальное число шагов и одновременно в пределах

длины записи промежуточных вычислений, не превосходящей некоторого полинома степени  $k$  от длины записи аргумента и использующих вспомогательные функции только из  $FP||n\text{-SPACE}$  и с древовидно-охраняемыми правилами;

б)  $FP||n^k\text{-SPACE}$ .

Доказательство теоремы 6 основано на моделировании вычисления каждым алгоритмом из 5) посредством машины Тьюринга и из 4) посредством алгоритма Маркова-Поста с вспомогательными функциями из  $FP||n\text{-SPACE}$  и с древовидно-охраняемыми правилами.

При этом число шагов и длина записи промежуточных вычислений по способу моделирования и условию теоремы оказываются с требуемыми верхними оценками.

Среди классов  $FP||n^k\text{-SPACE}$  (при различных  $k$ ) особенно важным для практических вычислений оказывается класс  $FP||n\text{-SPACE}$ , часто называемый  $FP||LIN\text{-SPACE}$ .

В качестве следствия можно привести пример предиката, принадлежащего классу  $FP||n^2\text{-SPACE}$ . Это существенно усиливает один из результатов работы [4].

Пусть  $Z_2$  – множество всех целых чисел, записанных в двоичной системе счисления,  $\bar{\quad}$  и  $\nabla$  – поразрядные отрицание и дизъюнкция,  $\|x\|$  обозначает длину записи числа  $x$ .

**Следствие.** Предикат проверки отрицательности постоянного терма в сигнатуре  $\langle Z_2; +, -, \cdot, =, \bar{\quad}, \nabla, \lambda x\|x\|, \lambda x\|x\|^2, \dots, \lambda x(x \cdot \|x\|^2), \lambda x(x \cdot \|x\|^3), \dots \rangle$  принадлежит классу  $FP||n^2\text{-SPACE}$ .

Действительно, можно построить алгоритм Маркова-Поста с охраняемыми правилами, который за линейное число шагов и с квадратичной памятью вычисляет предикат из формулировки следствия.

Отметим, что добавление функции  $\lambda x.x^2$  превращает предикат из следствия в NP-трудный [4]. В то же время после добавления новой функции объем функций, задаваемых термами с переменными, не увеличивается.

Представляют интерес также классы  $FP||n \log^c(n)\text{-SPACE}$ , объединение которых по всем целым  $c$  удобно обозначать посредством  $FP||QLIN\text{-SPACE}$ , то есть класс функций, вычислимых на машине Тьюринга, совершающей не более чем полиномиальное число шагов и использующей квазилинейную память.

**Теорема 7.** Пусть все вспомогательные функции принадлежат классу  $FP||QLIN\text{-SPACE}$ . Тогда имеет место последовательность включений следующих классов алгоритмов:

1) класс алгоритмов Маркова-Поста, вычислимых не более чем за квазилинейное число шагов и одновременно в пределах длины записи промежуточных вычислений, не превосходящей  $c_1 n \log^c(n)$  при некоторых  $c_1$  и  $c$ , где  $n$  – длина записи аргумента;

2) класс таких же, но рекурсивных алгоритмов, использующих только вспомогательные функции из  $FP||QLIN\text{-SPACE}$ , с теми же ограничениями и, кроме того, допускающих вычисление полиномов над длинами слов;

3) класс таких же алгоритмов с теми же ограничениями, но, возможно, ещё и с охраняемыми правилами;

4) класс таких же алгоритмов с теми же ограничениями, но, возможно, ещё и с древовидными правилами;

5) класс алгоритмов Маркова-Поста, вычислимых не более чем за полиномиальное число шагов и одновременно в пределах длины записи промежуточных вычислений, не превосходящей  $c_1 n \log^c(n)$  при некоторых  $c_1$  и  $c$  (где  $n$  – длина записи аргумента) и использующих вспомогательные функции только из  $FP||QLIN\text{-SPACE}$  и с древовидно-охраняемыми правилами;

б)  $FP||QLIN\text{-SPACE}$ .

Доказательство теоремы 7 основано на моделировании вычисления каждым алгоритмом из 5) посредством машины Тьюринга и из 4) посредством алгоритма Маркова-Поста с вспомогательными функциями из  $FP||QLIN\text{-SPACE}$  и с древовидно-охраняемыми правилами.

При этом число шагов и длина записи промежуточных вычислений по способу

моделирования и условию теоремы оказываются с требуемыми верхними оценками.

Теорема 7 важна для практических вычислений. В рамках класса  $\mathbf{FP}\|\mathbf{QLIN-SPACE}$  можно вычислить многие элементарные функции [8] над вещественными числами, приближения к которым можно быстро вычислять на машинах Тьюринга.

В случаях, когда сложно адекватно оценить степень полинома, ограничива-

ющего сверху число шагов работы алгоритма Маркова-Поста с дополнительными алгоритмическими средствами, оказываются полезными включения в каждой из в теорем 6 и 7 класса из п. 5 в класс из п. 6.

Кроме того, для класса из п. 5 теоремы 6 (и теоремы 7) справедлив совместный тезис Чёрча, полиномиальный по времени и ограниченный по памяти полиномом степени  $k$  (соответственно, квазилинейной памятью).

## Литература

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
2. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
3. Косовский Н.К. Условия полиномиальности алгоритмов, реализуемых на машинах Тьюринга с оракулами-функциями // XVII Международная школа-семинар «Синтез и сложность управляющих систем» имени академика О.Б. Лупанова (Новосибирск, 27 октября – 1 ноября 2008 г.). Новосибирск: Изд-во Института математики, 2008. С. 70–74.
4. Косовский Н.К., Косовская Т.М. Полиномиальность и NP-трудность задачи вычисления знака постоянного терма // Математические вопросы кибернетики. Вып.16. М.: Физматлит, 2007. С. 125–128.
5. Косовский Н.К., Косовская Т.М. Полиномиальный тезис Чёрча для рефал-5 функций, нормальных алгоритмов и их обобщений // Компьютерные инструменты в образовании, 2010. № 5. С. 12–21.
6. Мальцев А.И. Алгоритмы и рекурсивные функции. М.: Наука, 1986.
7. Турчин В.Ф. Рефал-5. Руководство по программированию и справочник // <http://www.refal.net/rf/5frm.htm>.
8. Яхонтов С.В. FLINSPACE конструктивные вещественные числа и функции // Saarbrücken: LAMBERT Academic Publishing, 2010.
9. Du D.Z., Ko K.I. Theory of Computational Complexity. A Wiley-Interscience Publication. John Wiley & Sons, Inc. 2000.

## Abstract

This paper contains two main sections. The first one (section 2) answers the question: «How to extend the notion of Markov algorithm by inclusion of a polynomial function of a word length and to conserve computation inside polynomial number of steps?». A sequence of mathematical notions of algorithm with more and more powerful computational tools is presented. It is proved that every of them coincides with the class of polynomial in time algorithms (class  $\mathbf{FP}$ ).

The second one (section 3) answers the question: «How to extend the notion of Markov algorithm in order to include it into subclass of  $\mathbf{FP}$  containing only algorithms using intermediate memory with the length not more than a polynomial of  $k$ -th degree?». The results of this section may be also regarded as a confirmation of the natural formulation of Church thesis for the last two classes of algorithms used in the formulations of the theorems of this section.

**Keywords:** polynomial time algorithms, class  $\mathbf{FP}$ , polynomial space algorithms, subclasses of  $\mathbf{FP-SPACE}$ , Markov algorithm, Post rules.

*Косовский Николай Кириллович,  
доктор физико-математических  
наук, профессор, заведующий  
кафедрой информатики математико-  
механического факультета СПбГУ,  
[kosov@NK1022.spb.edu](mailto:kosov@NK1022.spb.edu)*



Наши авторы, 2011.  
Our authors, 2011.