



ДОКАЗАТЕЛЬСТВА ПРИНАДЛЕЖНОСТИ РЕФАЛ-5 ФУНКЦИЙ НЕКОТОРЫМ ПОДКЛАССАМ FP-SPACE

Аннотация

Вводятся иерархии классов сложности функций как из **FP**, так и из **FP-SPACE**. В статье доказывается теорема о замкнутости классов функций этих иерархий, а также классов функций **FP||LIN-SPACE** и **FP||QLIN-SPACE** и классов предикатов **P||LIN-SPACE** и **P||QLIN-SPACE** относительно определения рефал-5 функций с ограниченными сверху числом рекурсивных вызовов и величиной использованной памяти.

Ключевые слова: машина Тьюринга, полиномиальное число шагов, рефал-5, подклассы класса **FP-SPACE**, подклассы класса **FP**.

ВВЕДЕНИЕ

По-видимому, класс полезных, результативных и эффективных для использования на компьютерах алгоритмов лежит внутри класса **FP-SPACE**. Одним из языков высокого уровня, предназначенных для решения задач искусственного интеллекта является язык рефал-5.

Доказательство возможности реализации рефал-5 функции на машине Тьюринга с полиномиальной (от длины исходных данных) верхней границей числа шагов вызывает существенное затруднение. Автором был доказан ряд теорем, условия которых обеспечивают такую возможность [3, 4, 5]. Отметим, что такая возможность устанавливает, по существу, полиномиальную по времени масштабируемость функций, удовлетворяющих условиям доказанных теорем. Ниже основная теорема из [3], по существу, распространяется на классы функций и предикатов, имеющих при реализации на машине Тьюринга как полиномиальные ограни-

чения по числу шагов, так и одновременно линейные (или квазилинейные), а также полиномиальные ограничения по памяти. Эти классы содержат многие функции, имеющие важное значение для практических вычислений [7].

Отметим, что рассматриваемые в работе классы замкнуты относительно операции подстановки, что является их весьма полезным математическим свойством.

1. ИСХОДНЫЕ ОПРЕДЕЛЕНИЯ

Определение. Класс **FP** – класс алгоритмов, вычисляемых на машинах Тьюринга [1, 8] за полиномиальное число шагов.

Определение. Функция называется квазилинейной, если она имеет вид $C_0((n+1) \lceil \log_2(n+2) \rceil^C + 1)$ при некоторых целых константах $C, C_0 \geq 0$.

Определение. При каждом целых $k, C \geq 0$ функция называется квазиполиномиальной степени (k, C) , если она имеет вид $C_0((n+1)^k \lceil \log_2(n+2) \rceil^C + 1)$ при некоторой целой константе $C_0 \geq 0$.

Квазилинейная функция является квази-полиномиальной степени $(1, C)$ при некотором C .

Определение. Класс **FLIN-SPACE** (соответственно **FQLIN-SPACE**, **F(n^k)-SPACE**, **FQ(n^k)-SPACE**, **FP-SPACE**) – это класс алгоритмов, вычислимых на машине Тьюринга с числом ячеек, ограниченным сверху какой-либо линейной (соответственно квазилинейной, полиномиальной степени k , квазиполиномиальной степени (k, C) , полиномиальной) функцией при некотором целом C .

Если в названии определённых здесь классов убрать букву **F**, то в соответствующих определениях слово *алгоритмов* должно быть заменено на слово *предикатов*.

Определение. Если **FT** и **FS** – классы сложности алгоритмов по числу шагов и используемой памяти машины Тьюринга соответственно, то посредством **FT||S** будем обозначать класс алгоритмов, для которых существует программа на машине Тьюринга с ограничениями из обоих классов одновременно. Аналогичный класс предикатов будем обозначать посредством **T||S**.

Ниже в работе будут рассматриваться классы функций **FP||LIN-SPACE**, **FP||QLIN-SPACE** и классы предикатов **P||LIN-SPACE** и **P||QLIN-SPACE**.

В [4] под шагом рефал-5 функции понималось успешное выполнение правила (то есть рефал-5 предложения) этой рефал-5 функции. Ниже сформулировано более детальное определение шага. Рассматриваются рефал-5 функции, записанные в виде программы на языке рефал-5 (см. также [5]).

В последовательности условий языка рефал-5 в качестве разделителя, как и в языке Пролог, используется запятая.

По сути дела, в языке рефал-5, в отличие от базисного рефала, в правилах может использоваться дерево условий, так как проверка выполнения каждого следующего условия зависит от успешности проверки предыдущего условия, а условие может содержать сравнение с несколькими образцами, обеспечивая, тем самым, ветвление.

Определение. Рефал-5 предложением (иначе говоря, правилом) называется образец с переменными для частей аргумента, за которым следует знак равенства и функциональное выражение или следует дерево условий, начинающееся с запятой, на листьях которого после знака равенства находятся функциональные выражения, возможно, использующие синтаксические (круглые) скобки, конкатенацию (приписывание друг к другу двух слов или функциональных выражений или вызовов функций с аргументами в виде функциональных выражений).

Рефал-5 функцией назовем следующее слово: её имя, после которого находится заключённая в фигурные скобки последовательность рефал-5 предложений, отделённых друг от друга точкой с запятой.

Иначе говоря, рефал-5 функция имеет следующий вид:
ИМЯФУНКЦИИ{ПРЕДЛОЖЕНИЕ₁;...;ПРЕДЛОЖЕНИЕ_N}.

Под выполненными рекурсивными вызовами рефал-5 функции f для аргумента Q будем понимать все вызовы этой функции, выполненные в процессе вычисления функции f для аргумента Q .

Рефал-5 функция может содержать вызовы вспомогательных (определённых ранее и встроённых) функций.

Введем определения нескольких характеристик вычислительной сложности для рефал-5 функций. Пусть список S содержит все вспомогательные функции, достаточные для вычисления рефал-5 функции f . В частности, метауниверсальная функция может вызывать только их.

Определение. Под числом рекурсивных вызовов при вычислении от аргумента Q рефал-5 функции f над списком функций S понимаем число выполненных рекурсивных вызовов функции f при вычислении значения $f(Q)$, измеряемое относительно длины аргумента Q (сложенной с длиной исходного содержимого стека).

В процессе вычисления рефал-5 функции от постоянного выражения возникает последовательность из выражений или наборов выражений (дополнительные выраже-

ния появляются для проверки условий (в том числе и вложенных) применения из очередной ветки условий применения очередного правила). При этом переменные в образцах для сравнения приобретают или меняют свои значения согласно (возможно, вложенным друг в друга) циклам по длине переменных для выражений. Вложенность определяется порядком первых вхождений в образец переменных для выражений. Кроме того, число выражений в каждом наборе ограничено сверху увеличенным на единицу максимальным числом условий в линейных ветвях правил.

Определение. Под шагом вычисления от аргумента Q рефал-5 функции f над списком функций S понимаем вычисление всех условий первой подходящей линейной (начинающейся с корня) ветви дерева условий в правилах определения функции f и всех предшествующих подветвей и ветвей, включая первые невыполненные условия во всех предшествующих правилах определения функции и предшествующих линейных ветвях дерева условий применения первого подходящего правила определения f .

Определение. Под длиной записи результата выполнения от аргумента Q рефал-5 предложения понимаем длину записи содержимого полученного стека, сложенную с длиной записи (возможно, функционального) выражения, находящегося на «листе» выполненной линейной ветви условий (дерева условий, начиная с образца сравнений, находящегося в его корне) и в которое вместо всех переменных из всех образцов и из выполненных вызовов подставлены их значения, полученные в результате сопоставления с образцом из корня этого дерева условий и проверки всех условий, находящихся на выполненной линейной ветви условий.

Определение. Под величиной использованной памяти при вычислении от аргумента Q рефал-5 функции f над списком функций S понимаем функцию от длины записи исходных данных, равную максимальной длине записи аргументов и результатов у всех выполненных рефал-5 предложений и рекурсивных вызовов f , каждый раз сложенной с длиной записи всего содержимого

стека в соответствующий момент шага вычисления вызова какой-либо необходимой функции. Эта величина измеряется относительно длины Q , сложенной с длиной исходного содержимого стека.

Вообще говоря, список S , являющийся параметром приведенных определений, может быть пустым. В этом случае слова «над списком функций S » могут быть вычеркнуты как в этих определениях, так и в дальнейших формулировках.

Отметим, что для определения величины использованной памяти достаточно учитывать максимум из длины записи величины исходных данных (включая исходное содержимое стека) из использованной памяти при осуществлении проверок всех условий применения правил только команд работы над стеком и из длины записи результирующего выражения в первой подходящей линейной ветви дерева условий применения f , в которую вместо переменных подставлены их значения.

2. ОСНОВНЫЕ РЕЗУЛЬТАТЫ

Следующая теорема представляет собой объединённую формулировку двух теорем.

Теорема 1. Класс FP-SPACE и класс предикатов P-SPACE замкнуты относительно задания рефал-5 функции и предиката, вычисляющих с полиномиальным верхним ограничением на величину использованной памяти, если используются вспомогательные функции только из класса FP-SPACE.

Схема доказательства. Необходимо реализовать каждый шаг вычисления рефал-5 правила, который заключается в последовательном присвоении значения каждой переменной из образца для сравнения с одновременным занесением имени переменной и её значения в используемую память и в проверке всех условий, необходимых для вычисления очередной линейной ветви условий (начинающейся с корня дерева условий) очередного рефал-5 правила.

Реализация шага рефал-5 функции f обеспечивается формированием аргументов для сравнений с образцами. В результате

успешности сравнения с ними происходит присвоение значений переменным, содержащимся в образце, или замена этих значений согласно (возможно, вложенным друг в друга) циклам по длине значений тех переменных, которые являются переменными для выражений, или выполнение самого левого (среди самых вложенных) вызова встроеной или вспомогательной (то есть определённой ранее) рефал-5 функции от постоянного (после подстановки значений переменных вместо самих переменных) выражения, или непосредственное преобразование в очередное выражение согласно первому применимому правилу определения f .

Выполнение каждого шага сопровождается использованием памяти для записи содержимого стека и для указания границ переменных из образца для сравнения в непосредственно вычисляемом на этом шаге выражении, а также для формирования и записи выражений (в том числе и дополнительных), подлежащих вычислению, и, наконец, для вычисления самого левого (среди самых вложенных) вызова вспомогательной функции. Выполненные дополнительные выражения, ставшие постоянными и не содержащие вызовов функций, стираются после установления безуспешности применения правила и после успешной проверки всех условий ветки дерева условий применения правила.

Это можно реализовать на машине Тьюринга с полиномиальными ограничениями на использованную память.

Максимальную по всем шагам вычисления суммарную длину записей, появляющихся в процессе пошагового преобразования (включая, во-первых, запись исходного выражения вместе с записью исходного содержимого стека и, во-вторых, запись результата вычисления вместе с заключительным содержимым стека), можно оценить сверху функцией необходимого вида.

Действительно, поскольку память ограничена для каждого класса функцией своего вида, то длина аргумента любой вычисляемой функции (в том числе и любого её рекурсивного вызова) будет ограничена функцией того же вида. Все рассматриваемые классы замкнуты относительно подстановки. Значит, каждая функция любого рассмат-

риваемого класса ограничена функцией вида, соответствующего этому классу.

Осталось промоделировать каждый шаг вычисления рефал-5 функции (включая проверку всех необходимых условий в очередной линейной ветви вычисления очередного правила). Это можно выполнить в пределах памяти того же типа на машине Тьюринга. Описание схемы доказательства завершено.

Следующая теорема также представляет собой объединённую формулировку двух теорем.

Теорема 2. *Класс функций \mathbf{FP} и класс предикатов \mathbf{P} замкнуты относительно задания рефал-5 функции и предиката, вычисляющих за полиномиальное (от длины исходных данных) верхнее ограничение на число её рекурсивных вызовов и полиномиальным верхним ограничением на использованную память, если используются вспомогательные функции только из класса \mathbf{FP} .*

Схема доказательства аналогична схеме доказательства теоремы 1 с дополнительной проверкой того, что моделирование каждого шага можно выполнить одной и той же программой машины Тьюринга не только в пределах памяти того же вида, но и одновременно за полиномиальное от длины исходных данных (включая исходное содержимое стека) число шагов. Действительно, поскольку все вспомогательные функции принадлежат соответствующему классу функций и выполнение каждого шага любого вызова любой функции не выводит за границу желаемого класса. Это действительно так, поскольку все вспомогательные функции принадлежат классу \mathbf{FP} , и выполнение каждого шага любого вызова любой функции не выводит за пределы желаемого класса. Кроме того, необходимо отметить, что число шагов вычисления рефал-5 функции ограничено сверху полиномом от суммы длины исходных данных и длины исходного содержимого стека, так как число шагов всех вложенных друг в друга циклов, возникающих при использовании сравнений с образцами, является полиномиальным.

Две следующие теоремы представляют собой бесконечные серии формулировок теорем.

Теорема 3. Для всякого целого $k \geq 0$ класс функций $F(n^k)$ -SPACE (соответственно класс функций $F(Qn^k)$ -SPACE, классы предикатов (n^k) -SPACE и (Qn^k) -SPACE) замкнут относительно задания рефал-5 функции f , вычисляющей с полиномиальным (соответственно квазиполиномиальным, полиномиальным квазиполиномиальным) степени k (соответственно $(k, 0)$, k , $(k, 0)$) верхним ограничением на величину использованной памяти. Кроме этого, могут использоваться вспомогательные функции только из класса FLIN-SPACE (соответственно FQLIN-SPACE, FLIN-SPACE, FQLIN-SPACE).

Схема доказательства аналогична схеме доказательства теоремы 1. Поскольку вспомогательные функции принадлежат только классу FLIN-SPACE или только классу FQLIN-SPACE и эти классы замкнуты относительно подстановки, а величина используемой памяти ограничена сверху посредством $C_0((n+1)^k \lceil \log_2(n+2) \rceil^C + 1)$ при некоторых $C, C_0 \geq 0$, то не только длина аргумента самого левого (среди самых вложенных) вызова функции ограничена сверху, но и длина результата (как и величина использованной памяти) вычисления каждой функции ограничена сверху функцией того же вида. Таким образом, моделирование каждого шага можно выполнить программой машины Тьюринга в пределах памяти необходимого вида.

Теорема 4. Для всякого целого $k \geq 0$ класс функций $FP(n^k)$ -SPACE (соответственно класс функций $FP(Qn^k)$ -SPACE, классы предикатов $P(n^k)$ -SPACE и $P(Qn^k)$ -SPACE) замкнут относительно задания рефал-5 функции или предиката, вычисляющих не более чем с полиномиальным (от длины исходных данных) числом её рекурсивных вызовов и полиномиальным (соответственно квазиполиномиальным, полиномиальным, квазиполиномиальным) степени k (соответственно $(k, 0)$, k , $(k, 0)$) верхним ограничением на использованную память. Кроме этого, могут использоваться вспомогательные функции только из класса $FP\|\text{LIN-SPACE}$ (соответственно $FP\|\text{QLIN-SPACE}$,

$FP\|\text{LIN-SPACE}$, $FP\|\text{QLIN-SPACE}$).

Схема доказательства аналогична схеме доказательства теоремы 1 с дополнительной проверкой того, что моделирование каждого шага можно выполнить одной и той же программой машины Тьюринга не только в пределах памяти того же вида, но и одновременно за полиномиальное от длины исходных данных (включая исходное содержимое стека) число шагов.

Теоремы 1, 2, 3 и 4 допускают некоторое усиление для случая задания нескольких рефал-5 функций совместной рекурсией. Такое задание можно заменить определением одной функции, аргумент которой дополнительно начинается с номера определяемой функции.

Теорема 5. Каковы бы ни были целые положительные числа k, C , функция 2^f (где $f(x) = ((n+1)^k \lceil \log_2(n+2) \rceil^C + 1)$ при n равном длине x) не может быть вычислена на многоленточной машине Тьюринга с числом ячеек, меньшим квазиполиномиальной функции степени как $(k, C-1)$, так и $(k-1, C')$ ни при каком C' . Но функция 2^f может быть вычислена на многоленточной машине Тьюринга с числом ячеек, не превосходящим f , за полиномиальное число шагов от длины исходного аргумента.

Схема доказательства. Получение нижней границы памяти (то есть в виде квазиполиномиальной функции степени как $(k, C-1)$, так и $(k-1, C')$ ни при каком C') основано на том, что запись результата вычисления предлагаемого примера функции больше указанной в теореме нижней границы результата вычисления. Получение верхней границы памяти основано на моделировании процесса последовательного вычисления функции 2^f на многоленточной машине Тьюринга, а также на двух дополнительных оценках сложности (как полиномиальности верхней оценки числа шагов, так и требуемой оценке числа используемых ячеек).

Теорема 5 позволяет утверждать о строгости предложенных иерархий как для класса FP -SPACE, так и для класса FP .

ЗАКЛЮЧЕНИЕ

Условия теорем 1 и 3 обеспечивают доказательство вычислимости с помощью рефал-5 функций с величиной памяти, ограниченной сверху некоторым полиномом. Поскольку NP-полные задачи (например задача ВЫПОЛНИМОСТЬ [1, 2]) могут быть решены с помощью таких вычислений, то для практических вычислений большее значение имеют теорема 2 и теорема 4, которые обеспечивают доказательство того, что вычисления рефал-5 функций, удовлетворяющих условиям теорем 2 или 4, осуществляются одновременно и в классе FP.

Использование доказанных теорем позволяет обеспечить математическое обосно-

вание теоретической и практической ценности программ, написанных на языке рефал-5 и удовлетворяющих условиям теоремы 2 или теоремы 4.

Кроме того, отметим, что теоремы о получении верхних оценок сложности одновременно как по памяти, так и по числу выполненных макрошагов (рекурсивных вызовов), как правило, отсутствуют в математической литературе. В результате можно сказать, что в статье реализован один из подходов доказательства некоторых важных для практики свойств программ, а именно, полиномиальной заданной степени масштабируемости по памяти и одновременно полиномиальной заданной степени масштабируемости по числу шагов.

Литература

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
2. Бабаев И.О., Герасимов М.А., Косовский Н.К., Соловьев И.П. Интеллектуальное программирование. Турбо-Пролог и Рефал-5 на персональных компьютерах. СПб.: Изд. СПбГУ, 1992.
3. Косовский Н.К., Косовская Т.М. Полиномиальный тезис Чёрча для рефал-5 функций, нормальных алгоритмов и их обобщений // Компьютерные инструменты в образовании. № 5, 2010. С. 12–21.
4. Косовский Н.К. Тезис Чёрча для полиномиальных по времени рекурсивных алгоритмов над словами и их длинами // Компьютерные инструменты в образовании, 2011. № 1. С. 4–11.
5. Косовский Н.К. Алгоритмы Маркова-Турчина и доказательства полиномиальной эффективности программ на языке рефал-5 // Компьютерные инструменты в образовании. № 4, 2012. С. 41–49.
6. Турчин В.Ф. Рефал-5. Руководство по программированию и справочник. <http://www.refal.net/rf\5frm.htm> (дата обращения 20.05.13).
7. Яхонтов С.В., Косовский Н.К., Косовская Т.М. Эффективные по времени и памяти алгоритмические приближения чисел и функций. Учебное пособие. СПб: Изд-во СПб. ун-та, 2012.
8. *Du D.Z., Ko K.I.* Theory of Computational Complexity. A Wiley-Interscience Publication. John Wiley & Sons, Inc. 2000.

PROOFS OF REFAL-5 FUNCTION BELONGING TO SOME SUBCLASSES OF FP-SPACE

Abstract

Complexity classes hierarchies for functions from both FP and FP-SPACE classes are introduced. Theorems about closure of these function classes of hierarchies as well as the classes of functions FP||LIN-SPACE and FP||QLIN-SPACE and classes of predicates P||LIN-SPACE and P||QLIN-SPACE under the definition of Refal-5 functions with bounded numbers of recursive calls and the used memory size are proved.

Keywords: Turing machine, polynomial number of steps, Refal-5, subclasses of the class FP-SPACE, subclasses of the class FP.

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



Наши авторы, 2013.
Our authors, 2013.