

Секция 4

МАТЕМАТИЧЕСКИЕ ОСНОВЫ
ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

УДК 510.52+004.051

FPT-АЛГОРИТМЫ И ИХ КЛАССИФИКАЦИЯ
НА ОСНОВЕ ЭЛАСТИЧНОСТИ

В. В. Быкова

Идея параметризованного подхода к оценке сложности алгоритмов и задач была предложена в 90-х годах прошлого столетия [1]. За последние два десятилетия эта идея нашла применение в различных областях компьютерных наук, таких, как криптография, искусственный интеллект, вычислительная биология, теория баз данных [2]. Суть подхода заключается в рассмотрении так называемых параметризованных задач и параметризованных алгоритмов.

Параметризованная задача Π состоит в том, что для заданных языка $L(\Pi) \subseteq \Sigma^* \times \mathbb{N}$, где Σ — некоторый конечный алфавит и $\mathbb{N}$ — множество всех неотрицательных целых чисел, и пары $(I, k) \in \Sigma^* \times \mathbb{N}$ требуется определить, является ли (I, k) элементом $L(\Pi)$. Для алгоритма α , решающего данную задачу, (I, k) называют входом, I — его основной частью, $n = |I|$ — длиной входа и число k — параметром задачи. Алгоритм α называют параметризованным, если его вычислительная сложность, определяемая как количество времени исполнения, оценивается с точки зрения длины входа n и значения параметра k . Таким образом, функция вычислительной сложности параметризованного алгоритма есть некоторая функция двух переменных $t(n, k)$.

Параметризованная задача Π считается разрешимой с фиксированным параметром, или FPT-разрешимой (*Fixed-Parameter Tractable*), если она может быть решена некоторым параметризованным алгоритмом за время $t(n, k) = O(n^{O(1)} \cdot f(k))$ для функции f , зависящей только от параметра k . Класс всех разрешимых с фиксированным параметром задач обозначается FPT. Соответствующие параметризованные алгоритмы, решающие такие задачи, называют FPT-алгоритмами. Очевидно, что в FPT лежат все полиномиально разрешимые задачи. Однако наибольший интерес представляют FPT-разрешимые задачи, являющиеся NP-трудными. С теоретической точки зрения все эти задачи могут быть решены за полиномиальное время при каждом фиксированном значении параметра. Уже имеются сборники параметризованных задач, включая FPT-разрешимые задачи [3].

Теория параметризованной сложности развивается по нескольким направлениям: определение иерархии классов сложности параметризованных задач, установление условий FPT-разрешимости, выявление взаимосвязи между параметризованной сложностью и классами приближенных алгоритмов, развитие методов анализа и разработки параметризованных алгоритмов. В многочисленных работах по параметризованной теории сложности для анализа и сравнения алгоритмов применяются методы классической (одномерной) теории сложности [4]. Между тем классическая одномерность ограничивает глубину анализа параметризованных алгоритмов, для которых вычислительная сложность описывается функцией от двух переменных $t(n, k)$.

В настоящей работе предлагается мера вычислительной сложности алгоритма, с помощью которой можно исследовать темп роста функций многих переменных, анализировать степень влияния структуры входных данных параметризованного алгоритма на его вычислительную сложность, сравнивать между собой FPT-алгоритмы. Этой мерой является частная эластичность. Представленные результаты являются обобщением и расширением результатов автора, опубликованных в [5–7] применительно к одномерной теории сложности вычислений.

Под частной эластичностью $E_x(z)$ функции $z = z(x, y)$ по аргументу x понимается эластичность переменной z , которая рассматривается как функция только от x и при постоянных значениях y . Частная эластичность $E_x(z)$ связана с частной производной функции $z = z(x, y)$ по x соотношением $E_x(z) = z'_x \cdot x/z$. Аналогично $E_y(z) = z'_y \cdot y/z$.

В общем случае $E_x(z)$, $E_y(z)$ являются функциями, зависящими от двух аргументов x и y . Однако ситуация упрощается, если учесть тот факт, что для большинства параметризованных алгоритмов время выполнения алгоритма описывается функцией вида $z(x, y) = q(x)f(y)$, где $q(x)$ — количественная компонента, а $f(y)$ — параметрическая компонента этой функции. Для такой формы представления функции $z = z(x, y)$ частные эластичности $E_x(z)$, $E_y(z)$ вырождаются в обычные эластичности функции одного аргумента: $E_x(z) = E_x(q(x))$, $E_y(z) = E_y(f(y))$. Теперь $E_x(z)$ зависит лишь от x , а $E_y(z)$ — от y . Каждая из функций $q(x)$, $f(y)$ принадлежит одному сложностному классу множества $\mathfrak{K} = \{\text{SUBPOLY}, \text{POLY}, \text{SUBEXP}, \text{EXP}, \text{HYPEREXP}\}$ по подобающему аргументу [5]. Обозначим через $\mathfrak{F}_x$ класс сложности функции $z(x, y)$ по аргументу x , а через $\mathfrak{F}_y$ — класс сложности по аргументу y . Тогда всякому параметризованному алгоритму с функцией вычислительной сложности $z(x, y) = q(x)f(y)$ соответствует пара $(\mathfrak{F}_x, \mathfrak{F}_y) \in \mathfrak{K} \times \mathfrak{K}$, характеризующая сложность данного алгоритма как по длине входа x , так и значению параметра y . Таким образом, мы приходим к двумерной классификации параметризованных алгоритмов. При двумерном подходе более отчетливую и общую формулировку получает определение FPT-алгоритма: параметризованный алгоритм называется FPT-алгоритмом, если его вычислительная сложность задается функцией $z(x, y) = q(x)f(y)$, для которой $(\mathfrak{F}_x, \mathfrak{F}_y) \in \{\text{SUBPOLY}, \text{POLY}\} \times \mathfrak{K}$.

Введенная двумерная классификация FPT-алгоритмов не противоречит понятиям, используемым в настоящее время в параметризованной теории сложности применительно к FPT-алгоритмам, а лишь формально их уточняет и расширяет. Допустимо дальнейшее развитие предложенного подхода в части увеличения числа параметров и анализа других форм представления функций сложности.

Подробное изложение представленных результатов можно найти в [8].

ЛИТЕРАТУРА

1. Downey R. and Fellows M. Parameterized complexity. New York: Springer Verlag, 1999.
2. Flum J. and Grohe M. Parameterized complexity theory. Berlin; Heidelberg: Springer Verlag, 2006.
3. Cesati M. Compendium of parameterized problems. <http://bravo.ce.uniroma2.it/home/cesati/research/compendium> — 2006.
4. Niedermeier R. Invitation to fixed-parameter algorithms: Oxford Lecture series in mathematics and its applications. Oxford: University Press, 2006.
5. Быкова В. В. Метод распознавания классов алгоритмов на основе асимптотики эластичности функций сложности // Журн. СФУ. Математика и физика. 2009. № 2(1). С. 48–61.

6. Быкова В. В. Эластичность алгоритмов // Прикладная дискретная математика. 2010. №2(8). С. 87–95.
7. Bykova V. V. Complexity and elasticity of the computation // Proc. of the 3-rd IASTED International Multi-Conference on Automation, Control, and Information Technology (ACIT-CDA 2010). Anaheim-Calgary-Zurich: ACTA Press, 2010. P. 334–340.
8. Быкова В. В. FPT-алгоритмы и их классификация на основе эластичности // Прикладная дискретная математика. 2011. №2. С. 40–48.

УДК 519.682

О СВОЙСТВЕ ФОРМАЛЬНЫХ ЯЗЫКОВ НЕПОСРЕДСТВЕННО СОСТАВЛЯЮЩИХ

К. В. Сафонов, Д. А. Калугин-Балашов

В теории формальных грамматик словарь языка $X = \{x_1, \dots, x_n\}$ обычно называют терминальным множеством, тогда как нетерминальным называют конечное множество $Z = \{z_1, \dots, z_m\}$ вспомогательных символов, необходимых для задания грамматических правил (грамматики) данного языка. Для элементов этих множеств определены операции конкатенации и формальной суммы, приводящие к мономам и многочленам [1].

Если мономы и многочлены построены в соответствии с грамматическими правилами данного языка, то они интерпретируются как предложения и совокупности предложений. Рассмотрение совокупности всех грамматически правильных предложений приводит к необходимости изучать формальные степенные ряды от терминальных символов.

Граматики непосредственно составляющих (нс-грамматики) являются важным подклассом контекстно-зависимых грамматик (кз-грамматик); в то же время контекстно-свободные грамматики (кс-грамматики) являются частным случаем нс-грамматик.

Нс-грамматике сопоставима система символьных уравнений

$$\alpha_j^{k_j} z_j \beta_j^{k_j} = p_j^{k_j}(x, z), \quad j = 1, \dots, m, \quad k_j = 1, \dots, L_j. \quad (1)$$

Ее решением является совокупность $(\alpha_1^1 z_1^1(x) \beta_1^1, \dots, \alpha_m^{k_m} z_m^{k_m}(x) \beta_m^{k_m})$ формальных степенных рядов от терминальных переменных $x = (x_1, \dots, x_n)$, а ряды $\alpha_1^{k_j} z_1^{k_j}(x) \beta_j^{k_j}$ являются нс-языком, определяемым данной грамматикой. Мономы $\alpha_j^{k_j}$ и $\beta_j^{k_j}$ называются контекстом. Рассмотрим *вполне определенные* нс-грамматики, которые сопоставимы системам уравнений вида

$$\alpha_j z_j \beta_j = p_j(x, z), \quad j = 1, \dots, m. \quad (1')$$

Эффективным инструментом изучения решений таких систем уравнений является *коммутативный образ* этой системы — новая система уравнений, переменные в которой рассматриваются как коммутативные, например из поля комплексных чисел. Понятно, что формальные степенные ряды $(z_1(x), \dots, z_m(x))$, являющиеся решениями исходной системы (1'), переходят в степенные ряды, представляющие алгебраические функции. Коммутативный образ некоторого многочлена или ряда r будем обозначать как $ci(r)$ (*commutative image*) [2].

Произведем ряд замен вида $z_j' = \alpha_j z_j \beta_j$. Получим систему уравнений

$$z_j' = p_j(x, z), \quad j = 1, \dots, m. \quad (2)$$