УДК 510.57

А.В. Тюленев
кандидат физико-математических наук, доцент
С.К. Соболев
кандидат физико-математических наук, доцент
М.В. Тюленева
старший преподаватель
Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет), г. Москва, Россия

ИЗЛОЖЕНИЕ ПОНЯТИЯ ВЫЧИСЛИМОЙ ФУНКЦИИ В ТЕХНИЧЕСКОМ УНИВЕРСИТЕТЕ

Аннотация. Широко известно сравнение значимости понятия вычислимой функции со значимостью понятия натурального числа (Э. Пост). Однако традиции преподавания в технических вузах не предусматривают знакомства с основами теории вычислимых функций, что затрудняет изучение сложности алгоритмов студентами таких специальностей, как «Информационная безопасность», САПР, «Прикладная математика» и др.

На основе опыта работы со студентами с различной математической подготовкой можно рекомендовать начинать изложение этой темы с формализации понятия алгоритма. В качестве таких конструкций предлагаются машины Тьюринга и нормальные алгорифмы Маркова.

Изучение различных формализаций интуитивного понятия алгоритма, сравнение решений, полученных упомянутыми методами, помогает лучшему пониманию теории вычислимых функций и способствует формированию чёткого представления о том, что такое сложность вычислений.

Ключевые слова: машина Тьюринга; нормальный алгорифм Маркова; вычислимая функция.

DOI: 10.25206/2307-5430-2020-8-249-254

1. Машины Тьюринга

Сперва рассматриваются одноленточные машины Тьюринга (МТ) с бесконечной в обе стороны лентой с одной читающе-пишущей головкой.

Подробное описание таких МТ можно найти, например, в [1, гл.3].

Полезно сразу рассмотреть несколько простейших программ.

Пример. Прибавление единицы. Ленточный алфавит $\Sigma = \{|, \#\}$, множество состояний $Q=\{q_0,q_1,q_2\}$ (начальное состояние q_1, конечное q_0). Вначале на ленте записано подряд $k\geq 0$ символов |, головка расположена в ближайшей пустой ячейке слева от слова. Программа может быть такой (знак ; разделяет команды):

$$q_1\# \to q_2\#R\ ;\ \ q_2| \to\ q_2|R\ ;\ \ q_2\# \to q_0|N.$$

После завершения работы головка указывает на крайний правый символ |.

Обратим внимание на то, что если попытаться обойтись без состояния q_2, используя вместо него q_1, то таблица сделается противоречивой. Если же отказаться от последней команды, то машина не останавливается, что часто ускользает при первом знакомстве с МТ.

Пример. Сложение. Ленточный алфавит Σ={|,+,#}, множество состояний $Q=\{q_0,q_1,q_2,q_3\}$. Вначале на ленте записано $k\geq 0$ символов |, потом +, затем ещё $n\geq 0$ символов |. Программа может быть такой:

$$q_1\# \to q_2\#R\ ;\ \ q_2| \to\ q_2|R\ ;\ q_2+ \to\ q_2|R\ ;\ q_2\# \to q_3\#L\ ;\ \ q_3| \to q_0\#L.$$

После завершения работы головка располагается на крайнем правом символе |, если хотя бы одно из слагаемых положительно. Типичной ошибкой при решении этой задачи является неявное предположение $kn>0$. Обратим также внимание на то, что стереть последний символ | можно только пройдя мимо него (иначе мы не знаем, последний ли он).

После приобретения некоторых навыков в построении МТ следует перейти к более сложным вопросам: чистым вычислениям и синтезу МТ.

Определение. МТ вычисляет (частичную) словарную функцию f *чисто*, если кроме результата $f(v)$ и пробелов на ленте ничего нет, а головка располагается непосредственно перед словом $f(v)$.

Теорема. Если словарная функция f вычисляется МТ M, то можно построить МТ M', которая чисто вычисляет f.

Доказательство полезно разбить на части и предложить их в качестве упражнений.

1. Построить МТ, которая делает то же самое, что M, но вместо пробелов пишет новый символ ◊.

2. "Чистка мусора" слева, потом справа.

3. Возврат головки к началу слова.

Следующая теорема показывает возможность "синтеза" МТ (доказательство целесообразно предложить в качестве упражнения).

Теорема. Для любых двух словарных функций, вычислимых на МТ, можно построить МТ, вычисляющую их композицию.

Что касается условных ветвлений и циклов, то имеет место следующая теорема (доказательство громоздко).

Теорема. На МТ можно осуществить ветвление типа

$$\text{IF } f(w) = \Lambda \text{ THEN } g(w) \text{ ELSE } h(w)$$

и итерации типа

$$\text{WHILE } f(w) = \Lambda \ \text{ DO } g(w)$$

(через Λ обозначается пустое слово).

МТ оказались формализацией интуитивного представления об алгоритме. Тезис Тьюринга: "для любого конечного алфавита и любой вычислимой (программой какого-либо языка программирования) частичной функции существует МТ, вычисляющая эту функцию".

Это утверждение не является теоремой; оно выражает то, что все известные алгоритмы удаётся реализовать на МТ. Если указать язык программирования, то тезис становится теоремой, и её можно доказать.

При рассмотрении многоленточных МТ следует подчеркнуть, что все решаемые на них задачи можно решить и на одноленточной машине. Для этого пришлось бы вводить дополнительные символы и состояния, обеспечивающие проходы влево-вправо для поиска данных и записи результатов. Что требует большего числа шагов, но это увеличение полиномиально. Последний факт хорошо подтвердить примером сложения двух чисел в двоичной записи на трёх- и одноленточной МТ.

Подробнее о МТ см., например, [1, гл.3, гл.5].

2. Нормальные алгорифмы Маркова

Пусть **A** – некоторый непустой алфавит, не содержащий точку "·" и стрелку "→"; пусть *A, B* – слова в этом алфавите.

Определение. *Простая формула подстановки* есть слово вида

$$A \to B$$

Заключительная формула подстановки есть слово вида $A \to \cdot B$.

Слово *A* называется *левой частью*, а слово *B* – *правой частью* формулы подстановки.

Определение. Формула *F действует* на слово *W* в алфавите **A**, если левая часть формулы входит в *W*. *Результатом действия F* на *W* считается результат подстановки правой части *F* вместо первого вхождения левой части *F* в слово *W*.

Определение. Конечный набор формул $F_1 \ \dots \ F_n$ называется *схемой в алфавите A*.

Определение. Схема α *действует на слово W*, если хотя бы одна из формул схемы действует на *W*. *Результат действия* есть результат действия на *W* первой по порядку формулы, действующей на *W*.

Если формула простая, то действие схемы – *простое*, если формула заключительная, то и действие схемы называется *заключительным*.

Пусть α – схема в алфавите ***A***, *W* – слово в ***A***. Определим индуктивно процесс применения α к *W*.

1-й шаг. Если схема α не действует на *W*, то процесс останавливается, а слово *W* считается результатом процесса. Если схема α действует на *W*, то шаг состоит в построении результата этого действия.

(k + 1)-й шаг процесса делается, если сделан k-й шаг процесса и если схема α действовала просто на k-м шаге. Тогда (k + 1)-й шаг состоит в построении результата действия схемы на результат k-го шага. Если схема α действовала на k-м шаге заключительно или α не действует на результат k-го шага, то (k+1)-й шаг не делается, процесс останавливается, а его результатом считается результат k-го шага.

Определение. Так определённые процессы называются *нормальными алгорифмами Маркова (НАМ) в алфавите* ***A***.

НАМ является *нормальным алгорифмом Маркова над алфавитом* ***A***, если он является НАМ в некотором расширении этого алфавита.

Если процесс применения НАМ к некоторому слову заканчивается, то алгорифм *применим* к данному слову; в противном случае – *неприменим*. Заметим, что нередко студенты склонны называть неприменимостью алгорифма отсутствие его действия на данное слово.

Пример. 1. Пусть $\boldsymbol{A} = \{a,b\}$ – алфавит (запятая "," в него не входит). Схема $a \to b$ заменяет все буквы a любого слова в этом алфавите на b.

В том же алфавите схема $a \to\cdot b$ заменяет первую слева букву a любого слова (в этом алфавите) на b.

2. Прибавление единицы к числу в унарной записи. $\boldsymbol{A} = \{|\}$, входное слово – |...|. НАМ со схемой $\to\cdot\ |$ прибавляет единицу к данному числу (пустое слово, по определению, является началом любого слова).

3. $\boldsymbol{A} = \{|, +\}$. Слово |...|+|...| является входным. НАМ со схемой $+ \to\cdot$ осуществляет сложение этих чисел (здесь "+" заменяется пустым словом).

Полезно сравнить сложности решения задачи сложения двух чисел m и n в унарной записи с помощью МТ и НАМ.

Для любых двух нормальных алгорифмов можно построить третий, который является их композицией; также возможно осуществление ветвлений и итераций (как и для МТ).

Имеет место следующий тезис.

Принцип нормализации. Всякий алгоритм в алфавите ***A*** эквивалентен некоторому НАМ над ***A*** (т.е. они перерабатывают одинаковые слова в одинаковые).

Как и тезис Тьюринга-Чёрча, принцип нормализации не может быть доказан. Однако доказано, что если словарная функция вычислима на МТ, то она вычислима и с помощью НАМ и наоборот, см. [2], [3, Гл.II, §5], [4, Гл.5, §§1,2]

3. Рекурсивные функции

Оказывается, класс всех вычислимых с помощью МТ или НАМ функций является класс *частично-рекурсивных функций (ЧРФ)*. Определение ЧРФ, их свойства и примеры, см., например, в [1], [5]. При первоначальном знакомстве с примитивно-рекурсивными функциями полезно запрограммировать вычисление базисных функций

$$s(x) = x + 1;\; o(x) = 0;\; I_{\mathrm{m}}^{\mathrm{n}}(x_1, \dots , x_{\mathrm{n}}) = x_{\mathrm{m}}, \quad m = 1, \dots , n$$

на МТ или с помощью НАМ.

Для лучшего понимания операции рекурсии рекомендуются доказать, что следующие функции являются ПРФ.

0. Константы $C_n(x)=n$, функция $h(x,y,z) = y+1$.

1. $h_1(x,y) = x + y$

2. $h_2(x,y) = xy$

3. $h_3(x,y) = 2^y$

4. $h_4(x,y) = x^y$.

5. $y!$.

Понятие оператора минимизации часто усваивается с трудом; можно предложить нетрудные задачи и примеры, облегчающие этот процесс.

1. Частичная функция $g(x,y) \approx x\text{-}y$ (определена только при $x \geq y$) получается с помощью операции минимизации из $f(x,y,z) = |x\text{-}(z+y)|$

2. Доказать, что $f(x,y)\approx x/y$ – ЧРФ (f определена, если и только если x делится на y).

3. Функция g может оказаться нигде не определённой: $g(x,y) \approx \mu y(s(x)+y = = 0)$.

Следующая теорема является кульминационным моментом.

Теорема. Функция вычислима на некоторой МТ, если и только если она частично-рекурсивна.

Доказательство см., например, в [4, Гл.3], [1, Гл.4], [5, Гл.11].

Библиографический список

1. Крупский В.Н., Плиско В.Е. Теория алгоритмов. – М.: Издательский центр "Академия", 2009. – 208 с.

2. Марков А.А. Элементы математической логики. – М.: Издательство Московского университета, 1984. – 80 с.

3. Марков А.А. Теория алгорифмов // Тр. МИАН СССР. Вып. 42. М.-Л.: Изд-во АН СССР, 1954. – С. 3 – 375.

4. Мендельсон Э. Введение в математическую логику. – М.: Наука, 1984. – 320 с.

5. Верещагин Н.К., Шень А. Вычислимые функции. – М.: МЦНМО, 2002. – 192 с.

Сведения об авторах:

Андрей Всеволодович Тюленев

Служебный почтовый адрес: Москва, 105005, Рубцовская набережная, 2/18, кафедра «Высшая математика»; e-mail: tyulenevav@ya.ru; spin-code: 6966-6032.

Научные интересы: методика преподавания высшей математики, обыкновенные дифференциальные уравнения, вычислимые функции.

Сергей Константинович Соболев

Служебный почтовый адрес: Москва, 105005, Рубцовская набережная, 2/18, кафедра «Высшая математика»; e-mail: sergeisob@bmstu.ru; spin-code: 4523-3619.

Научные интересы: методика преподавания высшей математики, автоматическая генерация заданий по математике для студентов.

Марина Валентиновна Тюленева

Служебный почтовый адрес: Москва, 105005, Рубцовская набережная, 2/18, кафедра «Высшая математика»; e-mail: tyulenevamv@ya.ru; spin-code: 5713 -0632.

Научные интересы: методика преподавания высшей математики, дискретная математика.