

ВЫЧИСЛИМОСТЬ ПО ТЬЮРИНГУ ЧАСТИЧНО РЕКУРСИВНЫХ ФУНКЦИЙ И МАТЕМАТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ О ХАНОЙСКОЙ БАШНЕ

Петренко Денис Александрович

студент, Самарский Университет им. С.П. Королёва, РФ, г. Самара

Семёнов Артем Витальевич

студент, Самарский Университет им. С.П. Королёва, РФ, г. Самара

Куликова Ирина Геннадьевна

научный руководитель, старший преподаватель ФГБОУ ВО «Самарский государственный университет путей сообщения», РФ, г. Самара

Аннотация. В статье рассматривается алгебраический способ доказательства вычислимости по Тьюрингу частично рекурсивных функций, уделяется внимание конструированию машин Тьюринга и выполнению операций над ними, уточняется понятие вычислимых функций. В дополнение разбирается вариант применения рекурсивных алгоритмов и преобразования их в нерекурсивные на примере рекурсивного решения популярной головоломки «Ханойская башня».

Ключевые слова: машина Тьюринга; рекурсия; рекурсивная функция.

Введение

Содержание деятельности по синтезу машин Тьюринга - достаточно непростая задача, включающая в себя выполнение действий по составлению соответствующих программ.

Рассмотрим пример. Пусть требуется смоделировать машину Тьюринга, способную из n расположенных подряд единиц выдавать на ленте $n - 2$ единицы, также расположенные подряд, если $n > 2$, и способную функционировать вечно, если $n = 0$ или $n = 1$.

Для успешного выполнения намеченного нам понадобится принять некоторые допущения. Далее по тексту будут иметь место следующие названия: так, за внешний алфавит мы примем двухэлементное множество $A = \{0, 1\}$. Число требуемых внутренних состояний предстоит вычислить в ходе написания программы. Принимаем допущение, что машина начинает функционировать из стандартного начального положения, которое определяется состоянием q_1 , и в котором обозревается крайняя правая единица из n размещенных на ленте.

Началом послужит стирание первой единицы, если она есть. Затем, переходя к следующей левой ячейке, сотрем там единицу, если, конечно, она в этой ячейке имеется. Каждый такой переход провоцирует машину на переход в некое новое внутреннее состояние, потому что в противном случае будут стерты вообще все единицы, записанные подряд. Тогда команды, отвечающие за озвученные действия, могут быть представлены в следующем виде:

$$q_11 \rightarrow q_20L; q_21 \rightarrow q_30L.$$

Далее пусть машина принимает состояние q_3 и обозревает третью справа ячейку из числа тех, где записано это слово (n единиц). Не внося изменения в содержательную часть обозреваемой ячейки, машина должна остановиться, или говоря иными словами, перейти в заключительное состояние q_0 , вне зависимости от содержимого ячейки. Тогда команды, отвечающие за озвученные действия, могут быть представлены в следующем виде:

$$q_3 0 \rightarrow q_0 0; q_3 1 \rightarrow q_0 1.$$

Представим себе случай, что на ленте записана только одна единица. Тогда, выполнив первый шаг, а именно команду $q_1 1 \rightarrow q_2 0Л$, машина перейдет в состоянии q_2 и будет обозревать вторую справа ячейку, где записан 0. Исходя из условия, в подобной ситуации машина должна функционировать вечно. Тогда команды, отвечающие за озвученные действия, могут быть представлены в следующем виде:

$$q_2 0 \rightarrow q_2 0П,$$

Далее, последовательно выполняя дополнительную команду шаг за шагом, машина движется по ленте бесконечно вправо (тот же эффект можно достичь, если протягивать ленту через считывающую головку справа налево). Представим случай, что на ленте не записано ни единой единицы, в этом случае машина, согласно условию, также должна функционировать вечно. И в этом случае в начальном состоянии q_1 обозревается ячейка с содержимым 0, а вечное функционирование машины достигается выполнением следующей командой:

$$q_1 0 \rightarrow q_1 0П$$

Представим разработанную программу (функциональную схему) смоделированной машины Тьюринга в виде таблицы:

Таблица 1.

Функциональная схема смоделированной машины Тьюринга

$A \backslash Q$	q_1	q_2	q_3
0	$q_1 0П$	$q_2 0П$	$q_0 0$
1	$q_0 0Л$	$q_2 0Л$	$q_0 1$

Подводя итоги, очевидно, что смоделированная машина Тьюринга может применяться к словам алфавита $A - \{0, 1\}$, состоящего из расположенных подряд n единиц ($n > 2$). Однако, с тем же успехом она может применяться к другим словам алфавита: 1011, 10011, 111011, 11011, 1100111, 1001111, 10111, 10110111, 10010111 и т.д.

Следует отметить, что сконструированная машина может быть не применима (говоря иными словами, при подаче этих слов на вход машины она функционирует вечно) к слову «1», а также к слову, состоящему из одних только нулей. Смоделированная машина не применима также к словам: 101, 1001, 11101, 101101, 1100101101 и т.д.

Разберем, что принято называть вычислимыми функциями. Ознакомимся с определением.

Определение. Функция называется вычислимой по Тьюрингу, если сконструирована машина Тьюринга, вычисляющая ее, иными словами машина, способная вычислять ее значения для тех наборов значений аргументов, для которых функция определена, и функционирующая вечно в случае, если функция для данного набора значений аргументов не определена.

Определение станет до конца точным, если принять определенные условности. Так, надо понимать, что разговор идет о функциях (может идти также и о частичных функциях, или

говоря иными словами, не всюду определенных), заданных на множестве натуральных чисел и принимающих натуральные значения.

Далее, необходимо принять некие условия, например, как записывать на ленте машины Тьюринга значения $x_1, x_2, \dots, x_n$ аргументов функции $f(x_1, x_2, \dots, x_n)$, из какого конкретно положения следует начинать переработку исходного слова, а также в каком конкретно положении предстоит снимать значение функции.

Данные условия можно обеспечить таким образом. Значения $x_1, x_2, \dots, x_n$ аргументов необходимо расположить на ленте в форме следующего слова:

01...101...10...01...10.
 $x_1 \quad x_2 \quad x_n$

Предстоит также принять некоторые обозначения. Так, натуральное x обозначаем:

$$\underline{1^x} = \underbrace{1 \dots 1}_x, \quad \underline{0^x} = \underbrace{0 \dots 0}_x.$$

Принимаем условие, что $0^0 = 1^0 = \Lambda$ — пустое слово. Тогда слова $1^0 = \Lambda$, $1^1 = 1$, $1^2 = 11$, $1^3 = 111$, ... рассматриваются как «изображения» натуральных чисел 0, 1, 2, 3, ... соответственно.

Тогда, предыдущее слово можно представить так: $01^{x_1}01^{x_2}0\dots01^{x_n}0$.

Очевидно, что приступить к переработке данного слова предстоит из стандартного начального положения, иными словами, из положения, когда в состоянии q_1 обозревается крайняя правая единица записанного слова.

Допустим, функция $f(x_1, x_2, \dots, x_n)$ определена на некотором наборе значений аргументов, тогда на ленте будет обозначено подряд $f(x_1, x_2, \dots, x_n)$ единиц; в противном случае машина должна будет работать бесконечно. И тогда при соблюдении, всех перечисленных условий, можно утверждать, что машина Тьюринга вычисляет данную функцию. Мы добились того, что сформулированное выше определение представлено как абсолютно верное.

Приведем пример: требуется смоделировать машину Тьюринга, вычисляющую функцию $f(x) = x/2$, которая не всюду определена: область ее определения - множество четных чисел. Учитывая определение, необходимо смоделировать машину Тьюринга, способную при подаче на ее вход четного числа выдавать на выходе половину этого числа, а при подаче нечетного - функционировать бесконечно долго.

Разберем вопрос о вычислимости по Тьюрингу частично рекурсивных функций.

Докажем следующую теорему:

Теорема. Если функция $f(x, y)$ правильно вычислима на машине Тьюринга, то и функция $\varphi(x) = \mu y[f(x, y) = 0]$, получающаяся с помощью оператора минимизации из функции $f(x, y)$, также правильно вычислима на машине Тьюринга.

Доказательство. Для успешного выполнения намеченного нам понадобится принять ряд обозначений. Допустим, что F — машина Тьюринга, правильно вычисляющая функцию $f(x, y)$. С ее помощью смоделируем машину Тьюринга, способную для заданного значения x вычислять поочередно значения $f(x, 0)$, $f(x, 1)$, $f(x, 2)$, ... до искомого момента, когда впервые появится $f(x, i) = 0$. По окончании данного события на ленте появится результат — число i , являющее собой значение функции $\varphi(x) = i$. Если выполняется условие, что для всех значений i справедлива функция $f(x, i) > 0$, машина будет функционировать вечно, и это обстоятельство станет свидетельством того, что функция φ не определена в точке x . Пусть начальная конфигурация на машине представлена таким образом: $q_1 0 1^x 0$. Представим ее в виде $q_1 0 1^x 0^0$

и применим к ней “копирование” K_2 . Получим конфигурацию $01^x 01^0 q_1 01^x 01^0$. Вычисляем значение $f(x,0)$, используя возможности машины

$$F: 01^x 01^0 q_\alpha 01^{f(x,0)}.$$

Переходим к следующим этапам. Далее необходимо подобрать команды, способные при условии $f(x,i) > 0$ переводить конфигурацию $01^x 01^i q 01^{f(x,i)}$ в конфигурацию $01^x 01^{i+1} q 01^{f(x,i+1)}$:

$$\begin{aligned} q_\alpha 0 &\rightarrow q_{\alpha+1} 0 \text{ П: } 01^x 01^0 0 q_{\alpha+1} 1^{f(x,0)}; \\ q_{\alpha+1} 1 &\rightarrow q_{\alpha+2} 0: 01^x 01^0 0 q_{\alpha+2} 1^{f(x,0)-1}; \\ q_{\alpha+2} 0 &\rightarrow q_{\alpha+3} 0 \text{ Л: } 01^x 01^0 q_{\alpha+3} 0 0 1^{f(x,0)-1}; \\ q_{\alpha+3} 0 &\rightarrow q_{\alpha+4} 1: 01^x 01^0 q_{\alpha+4} 1 0 1^{f(x,0)-1}; \\ q_{\alpha+4} 1 &\rightarrow q_{\alpha+5} 1 \text{ П: } 01^x 01^1 0 q_{\alpha+5} 0 1^{f(x,0)-1}; \\ O: 01^x 01^1 q 0; \\ (B^-)^2: 01^x 01^1 q 0 1^x 01^1; \\ F: 01^x 01^1 q_\beta 0 1^{f(x,1)}; \\ q_\beta 0 &\rightarrow q_\alpha 0: 01^x 01^1 q_\alpha 0 1^{f(x,1)}; \end{aligned}$$

Становится очевидным, что последняя команда завершает цикл программы, и машина от конфигурации $01^x 01^1 q_\alpha 0 1^{f(x,1)}$ принимает конфигурацию $01^x 01^2 q_\alpha 0 1^{f(x,2)}$, далее конфигурацию $01^x 01^3 q_\alpha 0 1^{f(x,3)}$ и так далее, по порядку. Рассмотрим случай, когда на определенном шаге машина принимает конфигурацию $01^x 01^i q_\alpha 0 1^{f(x,i)}$, при которой $f(x,i) = 0$. Это может означать, что $\varphi(x) = i$, в этом случае машина обязана выдать такой результат. Число i накоплено в «счетчике» 01^i . Значит, далее следует действовать согласно предлагаемому алгоритму. Имеющаяся команда $q_\alpha 0 \rightarrow q_{\alpha+1} 0 \text{ П}$ приведет машину к нужной конфигурации $01^x 01^i 0 q_{\alpha+1} 0$. Все последующие команды станут выдавать на ленту нужную конфигурацию $q_0 0 1^i$, говоря другими словами, $q_0 0 1^{\varphi(x)}$:

$$\begin{aligned} q_{\alpha+1} 0 &\rightarrow q_\gamma 0: 01^x 01^i 0 q_\gamma 0; \\ (B^-)^2: 01^x q_\gamma 0 1^i 0; \\ \text{ВО } B^-: q_\delta 0 1^i; \\ q_\delta 0 &\rightarrow q_0 0: q_0 0 1^{\varphi(x)} \end{aligned}$$

Теорема доказана.

В завершении предлагается рассмотреть возможности применения рекурсивных алгоритмов и преобразования их к нерекурсивным на примере решения задачи из информатики, основанной на красивой легенде о Ханойской башне

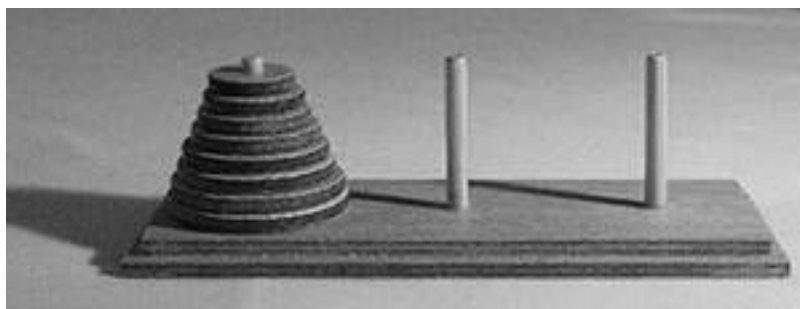


Рисунок 1. Модель Ханойской башни

Речь идет о некогда очень популярной головоломке XIX века. Ее изобрел французский математик Эдуард Люка в 1883 году.

Головоломка состоит из трех стержней, на один из которых надеты восемь колец. Все восемь колец различного размера и лежат меньшее на большем. Головоломка заключается в том, чтобы переместить все кольца на соседний стержень. При этом требуется выполнить минимальное количество ходов. За один ход разрешается перемещать лишь одно кольцо. Важным условием является также и то, что запрещается располагать большее кольцо на меньшем.

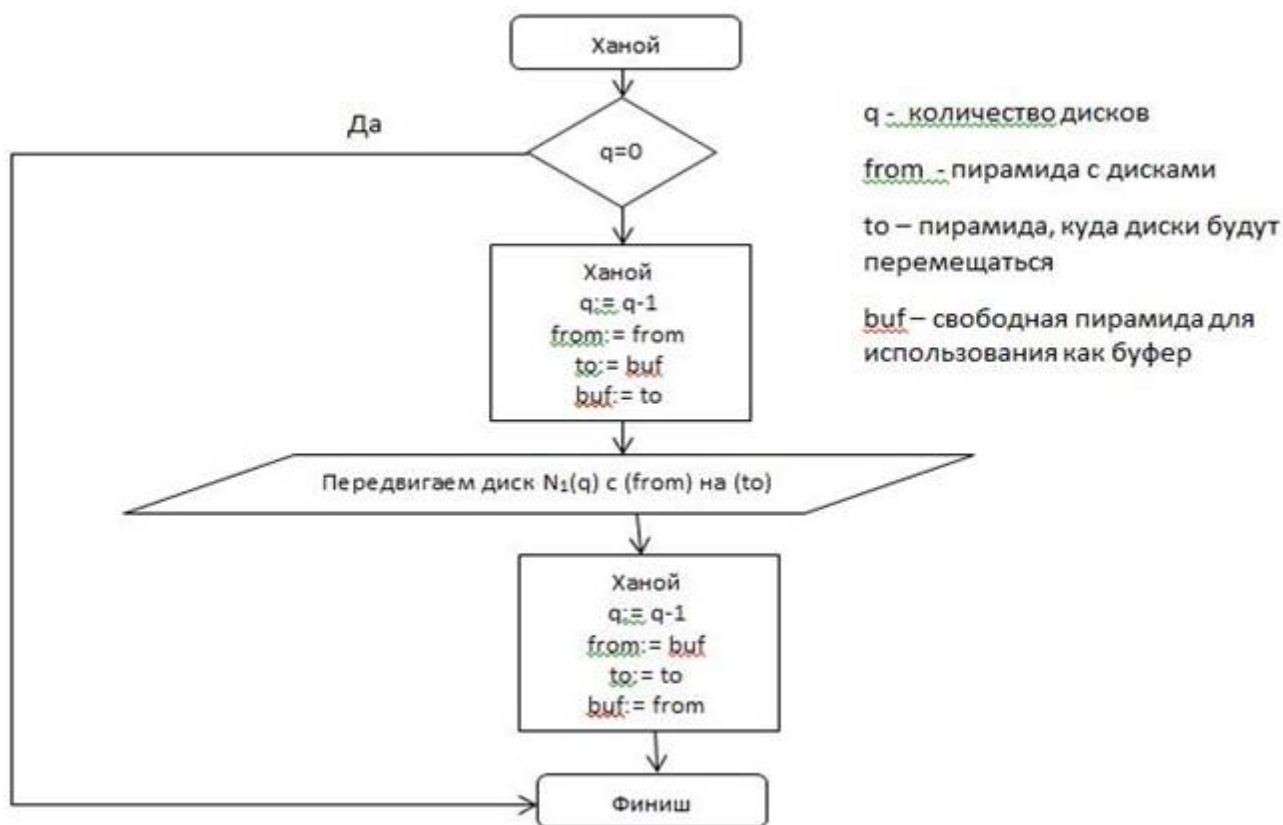


Рисунок 2. Блок-схема рекурсивного алгоритма решения

Известно некоторое количество решению данной задачи, и, несмотря на различные алгоритмы решений, результаты их одинаковые.

Более пристальному вниманию подлежит рекурсивное решение. Рассмотрим его.

Рекурсивно решаем задачу «переложить пирамиду из $n-1$ кольца на 2-й стержень». Затем перекладываем наибольшее кольцо на 3-й стержень, и рекурсивно решаем задачу «переложить пирамиду из $n-1$ кольца на 3-й стержень».

Далее методом математической индукции делаем вывод, что наименьшее количество ходов, требуемое для решения задачи, определяется как $2^n - 1$, где n — число стержней.

В информатике достаточно большое внимание уделяется решению задачи, построенной на легенде о Ханойской башне, где она является примером преобразования рекурсивных алгоритмов к нерекурсивным.

Что же касается самой легенде, то согласно ей, очень давно, где то в колыбели мира в Великом храме города Бенарес, под монастырем, отмечающим середину мира, был расположен диск из бронзы. На диске установлены три высоких алмазных стержня. Когда то монахи монастыря своим неповиновением заслужили негодование бога Брахмы. Тогда бог Брахма установил три высоченных стержня, возложил на единственный из них 64 диска, выполненных из чистого золота. Каждый меньший диск располагался на большем. И возвестил, что когда все 64 диска будут перемещены на соседний стержень, башня и храм станут пылью, и со страшными ударами молнии мир погибнет. Когда же это может произойти? Применяя способ рекурсивного решения можно найти решение. Количество перекладываемых дисков, которое предстоит выполнить монахам, исчисляется как 18 446 744 073 709 551 615.

Таким образом, даже при условии, что работать монахи станут день и ночь, перемещая каждую секунду по диску, их сложная работа заняла бы огромный период времени и длилась почти 585 миллиардов лет.

Заключение

В работе была рассмотрена и доказана вычислимость по Тьюрингу частично рекурсивных функций, а также рассмотрено математическое решение известной логической задачи.

Список литературы:

1. Носов В.А. Основы теории алгоритмов и анализа их сложности, 1992. - 140 с.
2. Ханойская башня (Савин А.) [Электронный ресурс]. - URL: <http://ipuzzles.ru/tower-of-hanoi/savin-tower-of-hanoi/> (дата обращения: 10.04.2019).