Лекция 16. Универсальная машина Тьюринга

Дискретная математика, ВШЭ, факультет компьютерных наук

(Осень 2014 – весна 2015)

Важнейшим свойством вычислимых функций является существование универсальной вычислимой функции. Выбрав в качестве формального определения алгоритма машины Тьюринга, мы получаем класс функций, вычислимых машинами Тьюринга. Нужно доказать, что этот класс функций обладает желаемыми свойствами. В частности, нужно доказательство существования универсальной машины.

Используя неформальное обоснование тезиса Чёрча – Тьюринга, легко убедить себя в том, что универсальная машина существует. Однако более детальное доказательство этого факта само по себе даёт дополнительную уверенность в тезисе Чёрча – Тьюринга и позволяет пересказывать результаты о неразрешимости, аналогичные тем, что были получены в абстрактной постановке.

Самый короткий путь к построению универсальной машины лежит через построение многоленточной универсальной машины. Как мы уже знаем, многоленточные машины моделируются одноленточными, так что этого достаточно.

1 Универсальная машина Тьюринга

По аналогии с абстрактной теорией хотелось бы сказать, что универсальная машина Тьюринга U моделирует работу любой другой машины в том смысле, что получив на вход описание МТ M и описание её входа x, она выдаёт тот же результат $M(x)$.

Однако такая машина невозможна: ведь её алфавит конечен, а существуют МТ со сколь большим алфавитом. Поэтому нужно научиться описывать и МТ, и их входные слова в каком-то одном алфавите. После этого уже можно сформулировать требования к универсальной машине.

Описание машин и их входов в одном алфавите — несложная задача, хотя и требующая утомительных подробностей.

Зафиксируем формат описания МТ и входных слов. Будем считать, что и алфавит, и множество состояний — это отрезки натурального ряда:

$$A = \{0, 1, \dots, n\}, \quad Q = \{0, 1, \dots, k\},$$

причём для единообразия полагаем начальное состояние $q_0 = 0$ и пустой символ $\Lambda = 0$. Таблица переходов — это множество пятёрок вида

$$(\langle\text{старый символ}\rangle, \langle\text{старое состояние}\rangle, \langle\text{новый символ}\rangle, \langle\text{новое состояние}\rangle, \langle\text{команда движения}\rangle).$$

В команде движения будем считать, что 0 отвечает сдвигу влево, 2 — сдвигу вправо, а 1 означает, что головка должна остаться в прежнем положении.

Для простоты числа мы будем кодировать равномерно: записи всех символов имеют одинаковую длину:

$$\langle i\rangle = \underbrace{00\dots0}_{i\ \text{символов}}\,1\underbrace{00\dots00000}_{n-i\ \text{символов}}.$$

Строка таблицы переходов будет задаваться пятью кодами такого вида, разделёнными символом #.

Итак, описание $\langle M\rangle$ МТ M в таком формате выглядит как набор записей вида

$$\#\langle a\rangle\#\langle q\rangle\#\langle a'\rangle\#\langle q'\rangle\#\langle d\rangle\#\ .$$

Входное слово кодируется аналогично как последовательность кодов входящих в него символов:

$$\#\langle a_1\rangle\#\langle a_2\rangle\#\langle a_3\rangle\#\dots\#\langle a_n\rangle\#\ .$$

Пустое слово будем кодировать как $\#10^n\#$ (код пустого символа).

Таким образом, для описания МТ и их входов мы используем алфавит $\{0,1,\#\}$. В абстрактной теории алгоритмов мы рассматривали универсальную функцию $U(p,x)$ от пар чисел (программа вычисления функции одного аргумента, аргумент). Теперь нам удобнее рассматривать функции от пар слов в алфавите $\{0,1,\#\}$, что не изменяет абстрактной теории благодаря существованию вычислимой биекции между словами и натуральными числами, как это уже объяснялось раньше.

Универсальной будем называть функцию $U\colon \{0,1,\#\}\times\{0,1,\#\}\to\{0,1,\#\}$, значение которой на входах $(\langle M\rangle,\langle x\rangle)$, где $\langle M\rangle$ — описание машины M, а $\langle x\rangle$ — описание входного слова x, равно $\langle M(x)\rangle$.

Универсальная машина (УМТ) должна вычислять какую-нибудь универсальную функцию. Мы ещё не определили, как МТ вычисляет функции от нескольких аргументов. Наиболее естественный способ — разделять аргументы каким-нибудь специальным символом $\odot$.

Определение 1. МТ M *вычисляет* функцию $f\colon B^*\times B^*\to B^*$, если для каждой пары (u,v) из области определения функции f результат работы M на входе $u\odot v$ равен $f(u,v)$, а для каждой пары (u,v) не из области определения f машина M не останавливается на входе $u\odot v$.

Аналогично определяется вычисление функции от нескольких аргументов.

Замечание 1. Обратите внимание, что поведение универсальной функции определено только на входах, которые являются кодами машины и входного слова.

Если есть УМТ, которая вычисляет какую-нибудь универсальную функцию на таких входах, она также как-то работает и на остальных парах слов. Именно такая вычислимая функция и будет точным аналогом универсальной функции из абстрактной теории.

Другими словами, мы допускаем «нестандартное» кодирование некоторых алгоритмов. Это не создаёт проблему, хотя при желании можно построить вычислимые биекции между кодами машин и всеми словами в алфавите $\{0,1,\#\}$, а также между кодами входных слов и всеми словами в алфавите $\{0,1,\#\}$.

2 Универсальная 3-ленточная машина для 1-ленточных машин

Мы построим 3-ленточную машину, которая вычисляет универсальную функцию. Поскольку любую 3-ленточную машину можно преобразовать в 1-ленточную, получим отсюда существование искомой 1-ленточной МТ.

Для вычисления универсальной функции 3-ленточная УМТ U моделирует работу произвольной МТ на произвольном входе. Идея моделирования достаточно проста. На одной ленте (лента машины) машина U хранит описание моделируемой МТ M, на второй (лента состояния) держит описание текущего состояния и текущего символа, на третьей (рабочая лента, она же входная и лента результата) поддерживает описание ленты моделируемой машины. УМТ начинает работу, приводя начальную конфигурацию к указанному виду. После этого работа машины M моделируется такт за тактом.

Опишем более подробно эти действия. Все они основаны на МТ, которые копируют или переносят часть одной ленты на другую, а также МТ, которые сравнивают слова на двух лентах.

Задача 1. Постройте (многоленточные) МТ, которые выполняют следующие действия.

(a) «Переход к указателю»: машина двигает головку до тех пор, пока головка не оказывается над заданным символом (указателем) #. В терминах конфигураций это означает, что конфигурацию $\dots q_0 x\#\dots$ машина должна переводить в конфигурацию $\dots x q_1\#\dots$. Здесь q_0, q_1 — состояния МТ, а x — слово, которое не содержит указателя.

(Будет также использоваться и переход влево к указателю.)

(б) «Копирование»: машина копирует содержимое области одной ленты на другую ленту. В терминах конфигураций это означает, что машина переводит конфигурацию

$$(\dots q_0 \lhd u \rhd \dots\ ,\ \dots q_0\#\dots)$$

в конфигурацию

$$(\dots q_1 \lhd u \rhd \dots\ ,\ \dots q_1\#u\dots).$$

Здесь q_0, q_1 — состояния МТ, символы-ограничители $\lhd$, $\rhd$ задают область, которую нужно скопировать, $\#$ — указатель, а слово u не содержит ни указателя, ни ограничителей.

(Конфигурация двухленточной машины — это пара слов.)

(в) «Перенос»: машина переносит содержимое области одной ленты на другую ленту. В терминах конфигураций это означает, что машина переводит конфигурацию

$$(\dots q_0 u \rhd \dots\ ,\ \dots q_0\dots)$$

в конфигурацию

$$(\dots q_1 \rhd \dots\ ,\ \dots q_1 u\dots).$$

Здесь q_0, q_1 — состояния МТ, символы-ограничители $\lhd$, $\rhd$ задают область, которую нужно скопировать, $\#$ — указатель, а слово u не содержит ограничителя $\rhd$.

(г) «Сравнение слов»: машина сравнивает две области на двух лентах. В терминах конфигураций это означает, что машина переводит конфигурацию

$$(\dots q_0 \lhd u \rhd \dots\ ,\ \dots q_0 \lhd v \rhd \dots)$$

в конфигурацию

$$(\dots q' \lhd u \rhd \dots\ ,\ \dots q' \lhd v \rhd \dots),$$

где $q' = q_1$ если $u = v$, а в противном случае $q' = q_2$. Здесь q_0, q_1, q_2 — состояния МТ, символы-ограничители $\lhd$, $\rhd$ задают сравниваемые области, а слова u, v не содержат ограничителей.

Кроме того универсальная машина должна уметь считать до пяти (строки таблицы переходов — это пятёрки). Мы не указываем явно расширенного множества состояний, которое позволяет это сделать, см. общие замечания об использовании «оперативной памяти» в МТ.

Универсальная машина получается последовательным соединением трёх машин: (1) подготовительной, (2) моделирующей такт работы, (3) финальной.

Подготовительная машина переносит с входной ленты на ленту описания МТ часть входа до символа $\odot$ (см. задачу 1в). Символ $\odot$ удаляется на первой ленте и головка сдвигается вправо. Если описание машины пусто, то подготовительная машина останавливается без запуска машины, моделирующей такт работы.

Далее подготовительная машина помещает на ленту состояния код начального состояния 10^k. Для этого она использует второй код в первой строке таблицы переходов и машину из задачи 1в. Однако скопированное состояние может и не быть начальным (мы не оговаривали порядок строк в таблице переходов). Поэтому подготовительная машина пишет на первую позицию кода 1, а на все последующие позиции вплоть до символа $\#$ она записывает 0, чтобы на второй ленте и впрямь появился код начального состояния[1], после чего переходит влево до символа $\#$ (задача 1а). На ленте описания МТ подготовительная машина переводит головку в крайнее левое положение (т.е. переходит влево по указателю $\#$).

На том работа подготовительной машины завершается.

Корректность работы машины очевидна из сделанного описания за одним исключением. Подготовительная машина берётся определить результат работы *пустой машины* — те такой

[1] Построение такой вспомогательной машины оставляем читателю в качестве упражнения.

МТ, таблица переходов которой не определена ни для одной пары (символ, состояние). Из построения видно, что результат оказывается равным входу. Это соответствует работе пустой машины — та останавливается на первом же такте и не сдвигает головку, поэтому вычисляет тождественную функцию.

Машина, моделирующая такт работы. Подготовительная машина заканчивает работу в такой конфигурации: на первой ленте (описания МТ) и второй ленте (описание текущего состояния) головки находятся над самой левой непустой ячейкой (содержащей разделитель #), а на третьей ленте головка находится над разделителем перед кодом символа в текущей ячейке машины M.

Моделирование такта работы машины M, описание которой лежит на ленте описания МТ, будет начинаться и заканчиваться именно в таких конфигурациях.

Моделирование такта работы состоит в выполнении следующих шагов, которые, как и выше, используют МТ из задачи 1 и переходы между разделителями на заданное число позиций (итерация МТ из задачи 1а нужное количество раз, которое не превосходит 5):

(1) Перейти вправо до # по ленте состояния и скопировать на эту ленту код текущего символа.

(2) Найти строчку таблицы переходов, которая соответствует текущему состоянию и символу моделируемой машины. Для этого последовательно проверять пятёрки кодов на ленте описания МТ (это строчки таблицы переходов моделируемой машины M), для каждой пятёрки первые два кода сравнить с текущим символом и текущим состоянием соответственно (задача 1г применительно к рабочей ленте и ленте состояния).

   - В случае совпадения искомая строчка найдена, перейти влево на первый символ # этой строчки.
   - В случае несовпадения перейти вправо на первый символ # следующей строчки.

   Если совпадения не найдено, управление передаётся финальной машине. Иначе выполняются следующие действия:

(3) скопировать четвёртый код в пятёрке на ленту состояния;

(4) скопировать третий код в пятёрке на рабочую ленту;

(5) выполнить команду движения: переход влево или вправо до символа # на рабочей ленте; если сдвиг выходит за границы закодированного слова, т.е. машина встретила пустой символ машины U, то добавить на рабочую ленту код пустого символа аналогично тому, как это делает подготовительная машина с кодом начального состояния.

Чтобы убедиться в корректности такой машины (т.е. проверить, что конфигурация изменяется в соответствии с тактом работы машины M), достаточно сравнить формат описания МТ, данный выше с указанной последовательностью действий.

Финальная машина помечает текущую позицию, после чего ищет справа первый код пустого символа. Поскольку этот код 10^n, то сравнение символа с пустым выполняется без труда. Далее машина пишет пустой символ (машины U) и возвращается в помеченную исходную позицию и убирает метку.

Итак, мы построили 3-ленточную машину, вычисляющую универсальную функцию. Поскольку 3-ленточные машины моделируются на 1-ленточных, существует и 1-ленточная машина, вычисляющая универсальную функцию.

3 Соответствие между абстрактной теорией алгоритмов и МТ

Мы уже реализовали на машинах Тьюринга два самых важных свойства алгоритмов: доказали, что композиция вычислимых МТ функций вычислима и построили универсальную машину Тьюринга.

Но этого пока недостаточно, чтобы доказать неразрешимость следующей *проблемы остановки*: даны описание машины Тьюринга и её входа, нужно узнать, останавливается ли эта машина на этом входе.

Действительно, мы доказывали, что неразрешимо множество $H = \{n : U(n, n)$ определена$\}$. Сейчас под n нужно понимать слово в алфавите $\{0, 1, \#\}$. Заметьте, что построенная нами универсальная функция вполне возможно даёт какой-то результат и для слов, которые не являются описаниями МТ и их входов. Вдруг окажется так, что вся «трудность» диагонального множества H прячется именно в таких «паразитных» парах?

Задача 2. Какой результат даёт построенная выше УМТ на парах $\langle M \rangle \odot \langle x \rangle$, где алфавит M состоит из 10 символов, а слово x в алфавите из 5 символов?

Для преодоления этой трудности необходимо использовать эффективность выбранного нами способа кодирования МТ.

Задача 3. Докажите, что на машинах Тьюринга разрешимы следующие множества слов:
(а) слова вида $\langle M \rangle$, где M — машина Тьюринга;
(б) слова вида $\langle x \rangle$, где x — слово в некотором алфавите;
(в) слова вида $\langle M \rangle \odot \langle x \rangle$, где M — машина Тьюринга, x — входное слово для M.

Разрешимость множества слов X означает, что существует МТ, которая даёт результат 1 на словах из X и результат 0 на остальных словах.

Теорема 1. *Проблема остановки алгоритмически неразрешима.*

Доказательство. Используем результат предыдущей задачи. Подправим универсальную функцию: на паре (u, v) подправленная функция даёт результат $\langle M(x) \rangle$, если $u = \langle M \rangle$, $v = \langle x \rangle$, и не определена иначе. Вычислимость такой подправленной функции следует из разрешимости множества пар (описание МТ, описание входа МТ), при этом она остаётся универсальной.

Если бы проблема остановки МТ была разрешима, то было бы разрешимым и диагональное множество для такой подправленной функции: алгоритм разрешения на входе w проверяет, является ли $w \odot w$ описанием МТ и её входного слова; если да, то применяет алгоритм разрешения для проблемы остановки, а если нет, то говорит, что w не принадлежит диагональному множеству. $\square$

Впрочем, неразрешимость проблемы остановки можно доказать и без высокой теории, прямым применением диагонального рассуждения.

Задача 4. Выведите напрямую из тезиса Чёрча–Тьюринга неразрешимость проблемы остановки (без использования существования УМТ или абстрактной теории алгоритмов).

Опишем и общий способ сопоставить абстрактную теорию алгоритмов и построенную нами универсальную функцию для машин Тьюринга. Для этого упорядочим слова в алфавите $\{0, 1, \#\}$ по правилу: более короткое слово меньше более длинного, а слова одинаковой длины упорядочиваются лексикографически. Этот порядок линейный и фундированный.

Задача 5. Постройте МТ, которая по слову в алфавите $\{0, 1, \#\}$ находит следующее слово в указанном порядке.

Используя задачи 3 и 5, легко построить вычислимые на МТ нумерации как МТ, так и входных слов для этих машин. Это позволяет ограничиться только значениями универсальной для МТ функции только на парах (описание МТ, описание входного слова). Действительно, на машинах Тьюринга вычислима такая универсальная функция от двух натуральных аргументов n и k с натуральными значениями: найдём n-ую машину M_n в порядке описаний МТ и k-е слово x_k в порядке описания входных слов машины M_n, значением функции будет номер результата $M_n(x_k)$ в порядке описания входных слов машины M_n.

Задание алгоритмов машинами Тьюринга — это главная нумерация. Действительно, пусть функция двух аргументов $V(n, x)$ вычислима МТ M, т.е. на входе вида $n \odot x$ машина M даёт результат $V(n,x)$. Как построить алгоритм, который по n находит описание МТ, вычисляющей функцию $f_n(x) = V(n,x)$? Одной из машин, вычисляющих f_n, является такая, которая ко входу x дописывает слева разделитель $\odot$, а потом первый аргумент n; далее она выполняет действия машины, вычисляющей $V(n,x)$. Описание такой машины легко строится по входу n: нужно описать машину, которая пишет разделитель, а потом слово n символ за символом в обратном порядке; потом добавить к ней фиксированное описание машины M.

Задача 6. Докажите существование МТ, реализующей указанное выше преобразование.

При изучении главных нумераций мы активно использовали биекцию между парами чисел и числами.

Задача 7. Докажите, что существует МТ, которая вычисляет биекцию $c\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, где

$$c\colon (x,y)\mapsto \binom{x+y+1}{2}+y.$$

Числа заданы в унарной записи (т.е. n записывается как слово 1^n).

Как определить перечислимые множества, если в качестве алгоритмов используются машины Тьюринга? Для этого нужно определить машину Тьюринга, которая порождает список элементов множества. Будем считать, что такая перечисляющая машина имеет специальную ленту результата, куда записывает элементы списка через разделитель #. При этом перечисляющая машина сдвигает головку на ленте результата только вправо (такое ограничение легко выражается в терминах таблицы переходов).

Ещё мы использовали в абстрактной теории отладочную функцию и возможность запуска нескольких алгоритмов одновременно.

Задача 8. Докажите существование МТ, которая по входу $\langle M\rangle\#\langle x\rangle\#1^t$ проверяет, останавливается ли машина M на входе x за t шагов.

Параллельное исполнение нескольких МТ (заданного числа) легче всего смоделировать, воспользовавшись многоленточными машинами. Но возможно реализовать также и параллельное исполнение МТ на всех её входах, разбивая работу на стадии: первая стадия состоит в исполнении первого такта на первом входе, вторая — второго такта на первом входе и первого такта на втором входе и т.д.: на N-й стадии исполняется N-й такт на первом входе, $(N-1)$-й такт на втором входе, $\ldots$, первый такт на N-м входе.

Задача 9. Докажите существование (многоленточной) МТ, которая реализует указанный выше процесс.

Используя эти соответствия, можно перенести все доказанные в абстрактной теории утверждения на машины Тьюринга. Приведём несколько следствий:

Следствие 1 (теорема Райса)**.** *Для любого нетривиального свойства $\mathcal{A}$ функций алгоритмически неразрешима задача: по описанию МТ проверить, вычисляет ли эта МТ функцию, обладающую свойством $\mathcal{A}$.*

Следствие 2. *Неперечислимо множество описаний МТ, которые не останавливаются на любом входе.*

Следствие 3. *Существует МТ, которая на любом входе выдаёт в качестве результата своё описание.*