Занятие 23

В условиях задач $\langle M \rangle$, $\langle x \rangle$ означают соответственно описание машины Тьюринга и входного слова в том формате, который был введён на лекции (и написан в черновике учебника).

Задача 23.1. Докажите, что существуют МТ, разрешающие следующие множества слов:

a) слова вида $\langle x \rangle$, где x — слово в некотором алфавите;

b) слова вида $\langle M \rangle$, где M — машина Тьюринга.

Решение. То, что слово x является словом в некотором алфавите (обозначим его A_n) означает, что размер этого алфавита конечен и равен n. Однако, наперёд размер этого алфавита неизвестен. Если бы он был известен, то тогда его можно было бы зашить в оперативную память и облегчить задачу, однако произвольность размера алфавита является ключевым моментом в построении универсальной машины Тьюринга.

Напомним в чём состоит описание $\langle x \rangle$ входного слова x над алфавитом $A_n = \{0, 1, \ldots, n-1\}$. Код i-ой буквы алфавита имеет вид

$$\langle i \rangle = \underbrace{00\ldots0}_{i} 1 \underbrace{00\ldots0}_{n-i}.$$

Таким образом, троичный алфавит A_3 имеет коды букв $\langle 0 \rangle = 100$, $\langle 1 \rangle = 010$, $\langle 2 \rangle = 001$.

Пусть $x = x_1 x_2 \ldots x_m$, где x_k – k-ый символ слова x. Тогда код $\langle x \rangle$ представляет собой посимвольное кодирование каждой буквы с разделителем $\#$ между кодами букв:

$$\langle x \rangle = \#\langle x_1 \rangle \# \langle x_2 \rangle \# \ldots \# \langle x_k \rangle \# \ldots \# \langle x_m \rangle \#.$$

Задача состоит в том, чтобы проверить по слову w, состоящему из символов $0, 1, \#$, является ли это слово кодом некоторого слова x над алфавитом (заранее неоговоренного) размера n.

Сформулируем условия, которые необходимы и достаточны для того, чтобы слово w являлось кодом.

1. Слово w начинается и заканчивается символом $\#$.

2. Между любыми двумя соседними разделителям # находится одинаковое ненулевое количество символов.

3. Между любыми двумя соседними разделителям # находится ровно одна единица.

Существование машин Тьюринга, которые проверяют первое и третье условие очевидно. Для первого условия МТ проверяет находится ли головка в начальном состоянии над символом #, далее перемещает головку вправо до встречи первого пустого символа, после чего делает один шаг влево и проверяет, что оказавшийся под головкой символ равен #.

Как доказать существование МТ, не предъявляя таблицу переходов самой МТ? Вполне достаточно разбить описание алгоритма, реализуемого МТ на блоки, и описать блоки алгоритма так, что в случае если у вас возникнет интерес, вы сможете для каждого блока построить МТ, реализующую этот блок алгоритма.

С третьим условием справиться тоже не очень трудно. Соответствующая МТ перемещает головку между блоками $\#\ldots\#$, внутри каждого блока увеличивает счётчик (который можно зашить в состояния) в случае встречи 1 пока тот не достигнет 2 и на правой границе # проверяет, равен ли счётчик 1. Если да, то обнуляет счётчик и продолжает эту процедуру для следующего блока, если он есть, а встретив пустой символ Λ завершает работу с положительным результатом. В случае, если счётчик после обработки некоторого блока оказался равным 0 или 2, то МТ завершает работу с отрицательным результатом. Мы считаем, что 1 – это положительный результат, а 0 отрицательный. Для того, чтобы МТ завершила работу с результатом 1 или 0 МТ достаточно заменить символ под головкой на Λ, сместиться на одну ячейку влево, записать в ней результат и остановиться.

Доказать существование МТ, проверяющей условие 2 уже не так просто. Для проверки этого условия удобно воспользоваться двухленточной МТ и фактом эквивалентности многоленточных МТ однолентночной. Двухленточная МТ копирует первый блок на вторую ленту и проверяет при копировании, что длина блока не ноль (достаточно счётчика с двумя состояниями: «0» и «$\geqslant 1$»). После чего МТ возвращает головку второй ленты на первый разделитель #, а далее синхронно двигает головки по обеим лентам вперёд (после копирования головка на первой ленте как раз находится над вторым разделителем прямо перед вторым блоком). Как только на второй ленте достигнут #, МТ проверяет находится ли # под головкой на первой ленте. Если нет, то заканчивает работу с отрицательным результатом, а если да, то перемещает головку второй ленты на первый разделитель и продолжает проверять совпадение длины следующего блока с первым. В случае если совпадение было полным, пока на первой ленте головка не дошла до пустого символа Λ, МТ завершает работу с положительным результатом.

Перейдём ко второму пункту задачи. Описание $\langle M\rangle$ МТ M задано кодировкой таблицы переходов: подряд идущими словами вида

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

каждое из которых соответствует записи $\delta(a, q) = (a', q', d)$ в таблице переходов. Множество состояний конечно и кодируется той же кодировкой, что и алфавитные символы. То же относится и к кодировке переходов – каждый переход кодируется как элемент трёхэлементного множества. Напомним, что при кодировке разделители не накладываются: в кодовом слове нет двух разделителей подряд.

Таким образом, чтобы проверить, что слово w над алфавитом $0, 1, \#$ является кодировкой некоторой МТ, необходимо и достаточно проверить, что w начинается и заканчивается на $\#$, а также что при разбиении w на пятёрки по разделителям

$$\#w_1^1\#w_2^1\#w_3^1\#w_4^1\#w_5^1\#\ldots\#w_1^i\#w_2^i\#w_3^i\#w_4^i\#w_5^i\#\ldots\#w_1^m\#w_2^m\#w_3^m\#w_4^m\#w_5^m\#$$

- первые и третьи компоненты всех пятёрок являются кодами слов над одним и тем же алфавитом;
- вторые и четвёртые компоненты всех пятёрок являются кодами слов над одним и тем же алфавитом;
- пятые компоненты всех пятёрок являются кодом над троичным алфавитом;
- не может быть одновременных равенств $w_1^i = w_1^j$ и $w_2^i = w_2^j$ (при $i \neq j$)[1].

Проверка корректности кодировок осуществима аналогично первому пункту задачи. Для этого можно на дополнительную ленту скопировать соответствующие компоненты всех пятёрок, а затем проверить корректность кодировок тем же способом, что и в первом пункте. Проверить отсутствие одинаковых пар можно используя дополнительную ленту, на которую при проходе входной ленты слева направо будут записываться компоненты $\#w_1^i\#w_2^i\#$, при этом при переходе к j-ой пятёрке на входной ленте головка на дополнительной ленте перемещается на начало и идёт проверка на совпадение $\#w_1^j\#w_2^j\#$ со всеми парами на дополнительной ленте.

□

Задача 23.2. 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.

Решение. Модифицируем универсальную машину Тьюринга. Напомним, что универсальная МТ моделирует работу МТ M на входе x, храня на одной ленте описание МТ $\langle M\rangle$, на другой описание входа $\langle x\rangle$, а на третьей описание состояния $\langle q\rangle$. УМТ проверяет есть ли в описании $\langle M\rangle$ пятёрка, начинающаяся с $\#\langle x_i\rangle\#\langle q\rangle$ (головка на ленте-входа находится на разделителе перед $\langle x_i\rangle$). Если такое правило нашлось, то УМТ меняет состояние и символ на соответствующих лентах согласно этому правилу и осуществляет движение головки к коду $\langle x_i\rangle, \langle x_{i-1}\rangle$ или $\langle x_{i+1}\rangle$ в зависимости от правила.

Опишем МТ H, решающую задачу, модифицировав УМТ. Добавим к УМТ ещё одну ленту – тактовую ленту, на которой в начале работы МТ H перенесёт 1^t. После

[1] В противном случае получается, что в таблице переходов дважды встречается переход для пары (a, q).

каждого такта движения МТ H сдвигает головку на тактовой ленте на один шаг влево. В случае если моделируемая МТ M остановилась до того как на тактовой ленте был достигнут пустой символ Λ, МТ H завершает работу с результатом 1; если же Λ была достигнута, а M при этом не остановилась, то H завершает работу с результатом 0. □

Задача 23.3. Пусть имеется МТ M, которая вычисляет функцию $V(n, x)$ от двух переменных. Докажите, что существует такая МТ S, которая по входу n выдаёт как результат описание МТ M_n, вычисляющей функцию $f_n(x) = V(n, x)$.

Решение. Мы считаем, что аргументы n и x функции $V(n, x)$ подаются на вход M с разделителем: $n\#x$.

Заметим сначала, что для любого числа n существует МТ $ID_n(x)$, результат работы которой $n\#x$: она просто добавляет $n\#$ справа от x. Для удобства можно считать, что число n записано в унарной записи, тогда нетрудно построить МТ, которая двигается влево на шаг, ставит разделитель, затем производит n движений влево и ставит в каждой клетке 1.

Искомая МТ M_n является соединением машин ID_n и M. Опишем как строится соединение машин. Пусть множество состояния первой машины имеет размер m_1, а второй m_2. Тогда машина M_n будет иметь $m_1 + m_2$ состояния, причём первые m_1 состояний являются состояниями первой машины, а последние m_2 второй. Мы считаем, что машины имеют одинаковый алфавит A. Запишем сначала описание машины ID_n, а затем описание машины M с учётом перенумерации их состояний – теперь размер кода каждого состояния равен $m_1 + m_2$. Заметим, что МТ ID_n имеет одно конечное состояние q_f – добавим в описание M_n описание правил $\forall a \in A \in A : \delta(q_f, a) = (q_0^M, a)$, где q_0^M – начальное состояние МТ M. Процесс построения описания машины M_n окончен: действительно, сначала будут применены все правила МТ ID_n, после того как она завершит свою работу, управление перейдёт к машине M, благодаря добавленным правилам. □

Задача 23.4. Выведите неразрешимость проблемы остановки МТ напрямую из тезиса Чёрча–Тьюринга, не ссылаясь на существование универсальной машины или абстрактную теорию алгоритмов.

Решение. Проблема останова состоит в проверке остановки МТ M на входе x по описанию машины и входа $\langle M \rangle \odot \langle x \rangle$. Тезич Чёрча–Тьюринга гласит, что всякая вычислимая функция вычислима на машине Тьюринга. Теперь аргументами вычислимых функций являются слова над алфавитом $\{0, 1, \#\}$.

Идея решения следующая. Предположение о том, что проблема останова разрешима повлечёт за собой существование вычислимой функции H, которая разрешает эту проблему. Но тогда, существует и вычислимая функция A, которая будет проверять с помощью H, останавливается ли МТ, описание которой подано на вход A, и если да, то тогда сама функция A будет не определена, а если МТ не останавливается, то функция A будет определена. Это позволит нам прийти к противоречию: по тезису Чёрча-Тьюринга для A существует МТ, которая реализует вычислимую

функцию A, но тогда либо на описании этой МТ A должна быть определена, когда МТ не останавливается[2], либо МТ должна не останавливаться, когда A определена. Ни того, ни другого быть не может – это приведёт нас к противоречию. Это стандартный пример применения диагонального метода.

Перед изложением основной части решения, введём вспомогательную функцию. Функция $f(w)$ возвращает w, в случае если w не является описанием МТ. Если же $w = \langle M \rangle$, то $f(w) = \langle w \rangle_M$, то есть f тогда возвращает описание слова w в алфавите, над которым определена МТ M. Мы делаем это для того, чтобы машине M можно было подать на вход собственное описание.

Перейдём к решению. Будем доказывать от противного. Если проблема останова разрешима, то вычислима следующая функция.

$$H(w) = \begin{cases} \#, & \text{если } w \text{ не является описанием МТ и её входа;} \\ 1, & \text{если } w = \langle M \rangle \odot \langle x \rangle \text{ и } M \text{ останавливается на } x; \\ 0, & \text{если } w = \langle M \rangle \odot \langle x \rangle \text{ и } M \text{ не останавливается на } x. \end{cases}$$

Но если функция H вычислима, тогда следующая функция также вычислима.

$$A(w) = \begin{cases} \#, & \text{если } H(w \odot f(w)) = \#; \\ 0, & \text{если } H(w \odot f(w)) = 0. \end{cases}$$

В случае $H(w \odot w) = 1$ функция A не определена. Согласно тезису Чёрча-Тьюринга, для вычислимой функции A существует МТ M_A, которая её вычисляет. Следующие рассуждения приведут нас к противоречию. Какое значение функция A будет иметь на описании $\langle M_A \rangle$?

Допустим, что $A(\langle M_A \rangle) = 0$. Тогда, это означает, что $H(\langle M_A \rangle \odot f(\langle M_A \rangle)) = 0$, что в свою очередь влечёт, что МТ $\langle M_A \rangle$ не остановилась на собственном описании, но тогда значение $A(\langle M_A \rangle)$ должно быть не определено, а оно равно нулю – противоречие. Допустим, что A не определена на $\langle M_A \rangle$. Но тогда, МТ M_A не останавливается на собственном описании, а значит $H(\langle M_A \rangle \odot f(\langle M_A \rangle)) = 0$, что влечёт $A(\langle M_A \rangle) = 0$ – опять приходим к противоречию. Случай $A(\langle M_A \rangle) = \#$ невозможен, т.к. $\langle M_A \rangle$ корректное описание МТ. Значит, функция A невычислима, а следовательно и функция H невычислима. Из невычислимости H получаем, что проблема останова неразрешима.

□

Задача 23.5. Пусть $T(\langle M \rangle, \langle w \rangle)$ — номер того такта работы машины Тьюринга M на входе w, на котором головка в последний раз оказывается над пустым символом. (Если головка никогда не оказывается над пустым символом или оказывается над ним бесконечно много раз, функция T на паре $(\langle M \rangle, \langle w \rangle)$ не определена.) Вычислима ли функция $T(\langle M \rangle, \langle w \rangle)$?

[2]Напомним, что вычислимая функция на входе не определена тогда и только тогда, когда реализующая её МТ на этом входе не останавливается.

Примечание. Для упрощения рассуждений в решении этой задачи разрешается ссылаться на тезис Чёрча–Тьюринга.

Решение. На первый взгляд кажется, что функция T является вариацией функции, разрешающей проблему останова. Эта проблема неразрешима, поэтому разрешающая её функция невычислима. Однако, в отличии от функции H из предыдущего решения, функция T вычислима. Прежде всего заметим, что у МТ M есть три варианта поведения на слове w:

1) M останавливается на w;

2) M не останавливается на w и при этом головка оказывается над пустым символом Λ неограниченное число раз;

3) M не останавливается на w и при этом головка оказывается над пустым символом Λ лишь конечное число раз.

Если бы было только первых два случая, то годилось бы следующее решение: моделировать на УМТ работу M на w, храня при этом в памяти последний такт на котором головка M оказывалась над Λ. Если M остановилась, вывести этот такт. Эта процедура очевидно реализуется вычислимой функцией.

Модифицируем эту процедуру так, чтобы модификация учитывала ещё и третий случай. Заметим, что в третьем случае, начиная с некоторого момента, МТ M не выходит за пределы двух соседних Λ и работает только на ограниченном отрезке ленты: действительно, иначе бы Λ встречался бесконечное число раз. А значит конфигураций M при работе на входе w конечное число. Поэтому помимо номера такта последнего прохода через Λ процедура будет хранить в памяти все конфигурации МТ. Причём перед добавлением новой конфигурации в память, процедура будет проверять не встречалась ли эта конфигурация до этого. Если конфигурация оказалась повторяющейся, значит на конфигурациях есть период, поскольку переход от одной конфигурации к следующей за ней однозначен. А периодичность конфигураций в свою очередь означает, что МТ ушла в цикл и уже не остановится. В случае, если период найден, процедура проверяет проходит ли головка МТ через Λ на этом периоде и если да, то не останавливается (поскольку тогда пустой символ будет посещён бесконечное количество раз), а если нет, то возвращает номер последнего прохода через Λ.

□

Задача 23.6. *Клеточный автомат* задаётся всюду определенной функцией $C\colon A \times A \times A \to A$, где A — некоторое конечное множество (алфавит). Мы считаем, что в алфавите есть такой символ $\Lambda \in A$ (пустой символ), для которого $C(\Lambda, \Lambda, \Lambda) = \Lambda$.

Автомат работает на бесконечной в обе стороны ленте, ячейки которой заполнены символами из алфавита A. Состояние ленты изменяется по следующему правилу: если в момент времени $t-1$ в ячейках с номерами $i-1$, i, $i+1$ записаны символы x, y, z, то в момент времени t в ячейке с номером i записан символ $C(x, y, z)$. (Правило применяется ко всем ячейкам одновременно.)

Если в начальный момент на ленте записано слово $x \in (A \setminus \Lambda)^*$, а остальные ячейки пустые, а в некоторый момент времени состояние ленты не изменилось и на ленте написано слово $y \in (A \setminus \Lambda)^*$, а остальные ячейки пустые, то говорим, что клеточный автомат на входе x выдаёт результат y. В противном случае результат работы автомата не определён. Таким образом, автомат вычисляет функцию из слов в алфавите $(A \setminus \Lambda)^*$ в слова в том же алфавите.

a) Докажите, что для любого клеточного автомата C существует машина Тьюринга M_C, которая вычисляет ту же функцию, что и автомат C.

b) Докажите, что для любой машины Тьюринга M существует клеточный автомат C_M, который вычисляет ту же функцию, что и машина M.

Замечание. Автомат C не может вычислять функцию, вычисляемую МТ, в случае если алфавит входа МТ совпадает с алфавитом выхода (как принято в нашем курсе). Это невозможно, поскольку если в некоторый момент на ленте клеточного автомата появится слово u в алфавите входа, то результат работы автомата на нём будет точно такой же, как если бы автомат начал работу со слова u. Таким образом, автомат не останавливается при выводе слова $u = M(w)$ во входном алфавите, а продолжает моделировать работу МТ M на u как на входном слове. У этой коллизии есть два решения: первое – построить клеточный автомат, который получая на вход начальную конфигурацию МТ M на слове w возвращает конечную конфигурацию M на w, а второй – сделать вывод автомата в новом алфавите. Мы приводим решение для обоих подходов.

Решение. Опишем МТ M_C, моделирующую работу клеточного автомата C. Для начала M_C ограничивает входное слово разделителями $\#$ и смещает головку к левому разделителю. Таким образом, конфигурация МТ имеет вид

$$\Lambda q \# w_1 w_2 \dots w_n \# \Lambda.$$

Здесь w_i – i-ый символ слова w. Опишем как МТ реализует один такт работы клеточного автомата. Сначала МТ смещает левую и правую границу в случае, если после левого разделителя и соответственно перед последним не стоит Λ:

$$\Lambda q \# \Lambda w_1 w_2 \dots w_n \Lambda \# \Lambda.$$

Далее за один проход слева направа МТ реализует один такт работы клеточного автомата. В начале МТ обрабатывает блок Λw_1. МТ запоминает символ w_1, а в общем случае первый символ после Λ за $\#$, заменяет Λ на символ $a = C(\Lambda, \Lambda, w_1)$ и кладёт в память Λ. Далее МТ также обрабатывает блоки по два символа: запоминает их, меняет первый символ на значение функции C от символа из памяти, первого и второго символа, а первый символ кладёт в память. На примере $w_1 w_2$ это выглядит так: в памяти у M_C лежит Λ, МТ заменяет w_1 на $C(\Lambda, w_1, w_2)$, кладёт в память w_1 и переходит к следующему блоку – блоку $w_2 w_3$. Этот процесс длится до достижения блока $\Lambda\#$, в котором Λ обрабатывается аналогично обработке $\#\Lambda$.

Поскольку $C(\Lambda, \Lambda, \Lambda) = \Lambda$, то все клетки за разделителями не меняются. Однако, если после первого (перед последним) разделителя больше не стоит пустая клетка,

необходимо сдвинуть разделители. В случае, если ни одного изменения за моделирование такта автомата не произошло, МТ удаляет правый разделитель, возвращается к левому разделителю, удаляет его, смещается вправо к началу слова и возвращает результат работы автомата C.

Докажем теперь, что для любой МТ M существует клеточный автомат C_M, который моделирует её работу. Автомат C_M будет моделировать работу M на её конфигурациях. Так, входу w МТ M, будет соответствовать вход q_0w автомата C_M.

Опишем правила C_M, в зависимости от переходов M

- $\delta(a,q) = (a', q', \ \ 0)$, $\forall b \in A : C_M(q,a,b) = (q', a', b)$;
- $\delta(a,q) = (a', q', +1)$, $\forall b \in A : C_M(q,a,b) = (a', q', b)$;
- $\delta(a,q) = (a', q', -1)$, $\forall b \in A : C_M(b,q,a) = (q', b, a')$.

Если мы закончим на этом описание автомата C_M, то мы получим автомат, который переводит начальную конфигурацию M на w в конечную, если M останавливается на w, или же C_M как и M никогда не остановится на w. Однако, результатом работы M является содержимое ленты справа от головки до первого пустого символа. Чтобы не строить в этой задаче клеточный автомат, который отчистит с ленты всё, кроме результата M, воспользуемся леммой об уборке муссора и потребуем, чтобы после окончания работы M оставляла на ленте только результат. Тогда всякая конечная конфигурация M будет иметь вид $q_f v$, а значит, на ленте C_M будет записано $\ldots \Lambda q_f v \Lambda \ldots$.

Осталось для каждого конечного состояния q_f добавить к правилам C_M правила $\forall a \in A : C_M(\Lambda, q_f, a) = (\Lambda, \Lambda, a)$.

Но если мы хотим быть педантами до конца, осталось решить последнюю проблему: добиться того, чтобы на вход C_M подавалось w, а не q_0w. Эта проблема решается просто. Добавим к алфавиту A его копию $\bar{A}$, где все символы будут подчёркнутыми. И добавим правила

- $\forall a \in A : C_M(\Lambda, a, \Lambda) = (q_0, \bar{a}, \Lambda)$;
- $\forall a, b \in A : C_M(\Lambda, a, b) = (q_0, \bar{a}, \bar{b})$;
- $\forall \bar{a}, \bar{b} \in \bar{A}, \forall c \in A : C_M(\bar{a}, \bar{b}, c) = (\bar{a}, \bar{b}, \bar{c})$.

Первые правила нужны на случай, если на вход автомату подаётся ровно один символ. В остальных случаях процесс перехода к начальной конфигурации МТ осуществляется правилами из второго и третьего пункта. Осталось добавить все правила, описаные нами выше для конфигураций, но уже для алфавита $\bar{A}$.

□