Московский физико-технический институт
Факультет инноваций и высоких технологий
Математическая логика и теория алгоритмов, весна 2019
Машины Тьюринга: условия задач с указаниями и решениями

Машиной Тьюринга называется кортеж $\langle \Sigma, \Gamma, Q, q_1, q_0, \delta \rangle$, где Σ, Γ и Q суть конечные непустые множества, причём $\Sigma \subset \Gamma$ и $\Gamma \cap Q = \varnothing$, $q_0, q_1 \in Q$, $q_0 \neq q_1$, а $\delta\colon (Q \setminus \{q_0\}) \times \Gamma \to Q \times \Gamma \times \{L, N, R\}$. Σ называется входным алфавитом, Γ — ленточным алфавитом, Q — множеством состояний, а q_1 и q_0 — начальным и завершающим состояниями соответственно. Функцию перехода δ записывают в форме списка команд вида $q_i a_j \to q_r a_s D$. (Так как это функция, определённая на $(Q \setminus \{q_0\}) \times \Gamma$, для каждой пары из незавершающего состояния и символа ленточного алфавита есть ровно одна команда). Среди элементов Γ выделяют специальный символ $\#$ (бланк, пробел, пустой символ, обозначается также как _, $\Box$, ƀ), не входящий в множество Σ.

Если K — конечное множество, то через K^* обозначается множество всех конечных последовательностей элементов из этого множества (слов в алфавите K). Последовательность из нуля элементов обозначается ε (или Λ) и называется *пустым словом.* На последовательностях определена операция *конкатенации* (приписывания), обозначаемая символом $\cdot$. Часто этот символ опускается, запись AB означает конкатенацию слов A и B. Под записью A^n понимается конкатенация n экземпляров слова A.

1. Чему равняется A^0? A^1?
Ответ: $A^0 = \varepsilon$, $A^1 = A$.

Конфигурацией машины Тьюринга называется слово вида $AqaB$, где $A, B \in \Gamma^*$, $a \in \Gamma$, а $q \in Q$.

2. В каком алфавите записывается конфигурация?
Ответ: $\Gamma \cup Q$.

3. Покажите, что по конфигурации можно однозначно восстановить A, q. a и B.
Указание: в записи конфигурации есть только один символ из Q.

Вычислением на машине Тьюринга называется последовательность конфигураций $c_1, \ldots, c_k$, в которой любые две соседние конфигурации c_l и c_{l+1} соотносятся следующим образом:

- если $c_l = Aq_ia_jB$ и в программе встречается команда $q_ia_j \to q_ra_sN$, то $c_{l+1} = Aq_ra_sB$;
- если $c_l = Aq_ia_jB$ и в программе встречается команда $q_ia_j \to q_ra_sR$, то $c_{l+1} = Aa_sq_rB$ (если $B = \varepsilon$, то $c_{l+1} = Aa_sq_r\#$);
- если $c_l = Aaq_ia_jB$ и в программе встречается команда $q_ia_j \to q_ra_sL$, то $c_{l+1} = Aq_raa_sB$ (если $c_l = q_ia_jB$, то $c_{l+1} = q_r\#a_sB$).

4. Покажите, что вычисление можно полностью восстановить, зная начальную конфигурацию и число шагов. Покажите, что вычисление может содержать не более одной конфигурации, содержащей q_0.

Указание: каждая конфигурация однозначно определяет следующую, при этом после конфигурации с q_0 не следует ни одна.

Неформально работа машины Тьюринга описывается так: имеется бесконечная лента, разбитая на ячейки. В каждой ячейке записан символ из Γ. У машины имеется каретка, в каждый момент указывающая ровно на одну ячейку. За один такт машина считывает символ с ячейки, на которую указывает, и в зависимости от внутреннего состояния переходит в новое состояние, заменяет содержимое ячейки и сдвигается влево, вправо или остаётся на месте. Конфигурация $AqaB$ означает, что машина находится в состоянии q, указывает на ячейку с символом a, слева от этой ячейки записано A, справа — B. а все остальные ячейки заполнены $\#$.

Машина вычисляет (частично определённую) функцию $f\colon \Sigma^* \to \Sigma^*$, если:

а) Для всех x, на которых f определена, существует вычисление на этой машине, начинающееся с конфигурации q_1x и заканчивающееся конфигурацией $q_0f(x)$;

б) Для всех x, на которых f не определена, не существует вычисления на этой машине, начинающегося с конфигурации q_1x и заканчивающегося конфигурацией, содержащей q_0.

5. Покажите, что второе условие равносильно следующему: « Существует сколь угодно длинное вычисление, начинающееся с q_1x».

Указание: если вычисление заканчивается не в состоянии q_0, то его можно удлинить.

6. Пусть $\Sigma = \{0,1\}$. Опишите машины, вычисляющие функции: $f(x) = x$, $f(x) = 0$, нигде не определённую.

Ответы:

а) $q_10 \to q_00N$, $q_11 \to q_01N$, $q_1\# \to q_0\#N$

б) $q_10 \to q_1\#R$, $q_11 \to q_1\#R$, $q_1\# \to q_00N$

в) $q_10 \to q_10N$, $q_11 \to q_11N$, $q_1\# \to q_1\#N$

7. Пусть $\Sigma = \{1\}$. Опишите машины, вычисляющие функции: $f(1^n) = 1^{n+1}$, $f(1^n) = 1^{n-1}$, $f(1^n) = 1^{2n}$, $f(1^n) = 1^{n^2}$.

Ответы:

а) $q_11 \to q_11L$, $q_1\# \to q_01N$

б) $q_11 \to q_0\#R$, $q_1\# \to q_1\#N$ (машина не определена на пустом слове) в) $q_1 1 \to q_2\#R$, $q_2 1 \to q_2 1R$, $q_2\# \to q_3\#R$, $q_3 1 \to q_3 1R$, $q_3\# \to q_4 1R$, $q_4\# \to q_5 1L$, $q_5 1 \to q_5 1L$, $q_5\# \to q_6\#L$, $q_6 1 \to q_6 1L$, $q_6\# \to q_1\#R$, $q_1\# \to q_0\#R$ ($q_4 1$ недостижимо, можно что угодно написать).

8. Пусть $\Sigma = \{0, 1\}$. Опишите машины, вычисляющие функции: $f(x) = xx$, $f(x) = x\bar{x}$, где $\bar{x}$ — слово, полученное из x превращением нулей в единицы и наоборот, $f(x) = xx^R$, где x^R — слово x, записанное задом наперёд.

9. Пусть $\Sigma = \{0, 1\}$. Опишите машины, вычисляющие функции: $f(x) = 2x$, $f(x) = x + 1$, $f(x) = x - 1$, где x понимается как двоичная запись натурального числа.

Ответы:

а) $q_1 1 \to q_1 1R$, $q_1 0 \to q_1 0R$, $q_1\# \to q_2 0L$, $q_2 1 \to q_2 1L$, $q_2 0 \to q_2 0L$, $q_2\# \to q_0\#R$. (Это работает, если разрешить двоичной записи начинаться с нуля, в противном случае нужно отдельно проверять, что если вначале ноль, то дальше ничего нет, и тогда ничего не менять. А если есть, то можно зациклиться.)

б) $q_1 1 \to q_1 1R$, $q_1 0 \to q_1 0R$, $q_1\# \to q_2\#L$, $q_2 0 \to q_3 1L$, $q_2 1 \to q_2 0L$, $q_2\# \to q_0 1N$, $q_3 1 \to q_3 1L$, $q_3 0 \to q_3 0L$, $q_3\# \to q_0 1R$.

в) $q_1 1 \to q_1 1R$, $q_1 0 \to q_1 0R$, $q_1\# \to q_2\#L$, $q_2 1 \to q_3 0L$, $q_2 0 \to q_2 1L$, $q_2\# \to q_0\#R$, $q_3 1 \to q_3 1L$, $q_3 0 \to q_3 0L$, $q_3\# \to q_0 1R$. (Эта машина оставляет ноль нулём, если пытается вычесть 1).

Машина вычисляет (частично определённую) функцию $f\colon \Sigma^* \times \Sigma^* \to \Sigma^*$, если:

а) Для всех пар (x, y), на которых f определена, существует вычисление на этой машине, начинающееся с конфигурации $q_1 x\#y$ и заканчивающееся конфигурацией $q_0 f(x, y)$;

б) Для всех пар (x, y), на которых f не определена, не существует вычисления на этой машине, начинающегося с конфигурации $q_1 x\#y$ и заканчивающегося конфигурацией, содержащей q_0.

10. Пусть $\Sigma = \{0, 1\}$. Опишите машины, вычисляющие функции $f(x, y) = x + y$, $f(x, y) = x - y$, где x и y понимаются как двоичные записи натуральных чисел.

11. Опишите машину, вычисляющую $f(x, y) = xy$ (умножение чисел в двоичной записи).

Также рассматривают распознающие (разрешающие) машины, у которых вместо одного завершающего состояния q_0 есть два: принимающее q_a и отвергающее q_r.

12. Дайте формальное определение такой машины как кортежа.

Указание: кортеж должен быть таким: $(\Sigma, \Gamma, Q, q_1, q_a, q_r, \delta)$, а условия такие же, как и раньше.

Распознающая машина разрешает множество $A \subset \Sigma^*$, если:

а) Для всех x, лежащих в A, существует вычисление на этой машине, начинающееся с конфигурации q_1x и заканчивающееся конфигурацией, содержащей q_a;

б) Для всех x, не лежащих в A, существует вычисление на этой машине, начинающееся с конфигурации q_1x и заканчивающееся конфигурацией, содержащей q_r.

13. Пусть $\Sigma = \{1\}$. Опишите машины, распознающие множества: $\{1\}$, $\{1^n \mid n\vdots 2\}$, $\{1^{n^2} \mid n \in \mathbb{N}\}$.

Ответы:

а) $q_1\# \to q_r$, $q_11 \to q_21R$, $q_2\# \to q_a$, $q_21 \to q_r$.

б) $q_11 \to q_21R$, $q_21 \to q_11R$, $q_1\# \to q_a$, $q_2\# \to q_r$.

14. Пусть $\Sigma = \{0, 1\}$. Опишите машины, распознающие множества: $\{0\}$, $0^* = \{0^n \mid n \in \mathbb{N}\}$, $0^*1^* = \{0^n1^m \mid n, m \in \mathbb{N}\}$, $\{0^n1^n \mid n \in \mathbb{N}\}$.

15. Пусть $\Sigma = \{0, 1\}$. Опишите машину, распознающую множество палиндромов, т.е. $\mathsf{PAL} = \{x \mid x = x^R\}$.

Решение: Машина будет такой: $q_10 \to q_2\#R$, $q_11 \to q_3\#R$, $q_20 \to q_20R$, $q_21 \to q_21R$, $q_30 \to q_30R$, $q_31 \to q_31R$, $q_2\# \to q_4\#L$, $q_3\# \to q_5\#L$, $q_41 \to q_r$, $q_50 \to q_r$, $q_40 \to q_6\#L$, $q_51 \to q_6\#L$, $q_60 \to q_60L$, $q_61 \to q_61L$, $q_6\# \to q_1\#R$, $q_1\# \to q_a$, $q_4\# \to q_a$, $q_5\# \to q_a$.

Докажем формально, что она работает правильно. Для этого нужно воспользоваться индукцией. В качестве базы докажем, что машина принимает пустое слово и оба однобуквенных, а в качестве перехода докажем вот что: если слово начинается и заканчивается на разные буквы, т.е. $x = 0y1$ или $x = 1y0$, то машина возвращает ложь, а если слово имеет вид $0y0$ или $1y1$, то машина на нём возвращает тот же ответ, что и на входе y.

Действительно, пустое слово машина примет, т.к. у неё есть команда $q_1\# \to q_a$. Однобуквенное она сотрёт первой командой, например, $q_10 \to q_2\#R$, затем последовательно выполнит команды $q_2\# \to q_4\#L$ и $q_4\# \to q_a$ и тоже примет. Значит, база доказана.

Теперь докажем переход. Пусть например, $x = 0y1$. Тогда после первой команды машина находится в конфигурации q_2y1. Затем она будет двигаться вправо, ничего не меняя, и придёт в конфигурацию $y1q_2\#$. На следующем шаге получится конфигурация yq_41, а потом машина отвергнет вход. На входе вида $x = 1y0$ машина поведёт себя аналогично с заменой q_2 на q_3 и q_4 на q_5. Теперь пусть $x = 0y0$. Тогда машина первым шагом перейдёт в конфигурацию q_2y0, затем после ряда шагов в $y0q_2\#$ и потом в yq_40. После этого машина сотрёт последний ноль, перейдёт в состояние q_6 и будет двигаться налево, пока не придёт в конфигурацию $q_6\#y$. После этого она придёт в q_1y и потому вернёт тот же ответ. Случай $x = 1y1$ рассматривается аналогично.

16. Назовём временем работы машины максимальную длину вычисления на входах длины n. Докажите, что никакая машина, распознающая множество палиндромов, не может работать за время $o(n^2)$.

Решение. Докажем, что на некотором входе потребуется $\Omega(n^2)$ шагов. Для удобства будем рассматривать входы длины $4n$. Рассмотрим лишь слова вида $x0^{2n}y^{\mathtt{R}}$, где $|x| = |y| = n$. Слово такого вида является палиндромом тогда и только тогда, когда $x = y$. Неформально говоря, для проверки этого факта нужно перенести информацию об x (n битов) на расстояние хотя бы $2n$, что с учётом конечности памяти, заключённой во внутреннем состоянии, потребует $\Omega(n^2)$ операций. Теперь проведём формальное рассуждение.

Назовём протоколом работы машины в точке i список состояний и направлений, в которых машина пересекала границу между i-ой и $(i+1)$-ой ячейками.

Утверждение. Если $x \neq z$, то машина, распознающая язык PAL, имеет разные протоколы на $x0^{2n}x^{\mathtt{R}}$ и $z0^{2n}z^{\mathtt{R}}$ для любого $i \in [n, 3n]$.

Доказательство. Если машина работает правильно, то она принимает и $x0^{2n}x^{\mathtt{R}}$, и $z0^{2n}z^{\mathtt{R}}$. Докажем, что если протоколы в точке i одинаковы, то такой же протокол (и тот же результат) будет для $x0^{2n}z^{\mathtt{R}}$ и $z0^{2n}x^{\mathtt{R}}$, так что машина ошибётся на этих входах. Действительно, работа машины на замкнутом отрезке полностью описывается начальным состоянием на этом отрезке и протоколом работы на его границах. Если протоколы в точке i одинаковы, то машина слева от этой точки будет одинаково работать на всех входах, начинающихся с $x0^{i-n}$, а справа — на всех входах, оканчивающихся на $0^{3n-i}z^{\mathtt{R}}$. Значит, на входе $x0^{2n}z^{\mathtt{R}}$ левее точки i машина будет работать так же, как на входе $x0^{2n}x^{\mathtt{R}}$, а правее — так же, как на входе $z0^{2n}z^{\mathtt{R}}$. Поскольку и там, и там ответ положительный, то такой же ответ будет на $x0^{2n}z^{\mathtt{R}}$, т.е. машина сработает неправильно. Значит, протоколы должны быть разными.

Вернёмся к доказательству основного утверждения. Поскольку у машины Тьюринга конечное число состояний (пусть q), то любой протокол можно записать в алфавите из $2q$ символов (q состояний и 2 направления). Поскольку для всех 2^n слов $x \in \{0,1\}^n$ протоколы должны быть разными, то для любой точки i найдётся протокол длины $\log_{2q} 2^n - 1 = \Omega(n)$. (Если самый длинный протокол содержит l символов, то общее число протоколов не превосходит $1 + 2q + (2q)^2 + \cdots + (2q)^l$, что меньше $(2q)^{l+1}$. Отсюда $2^n < (2q)^{l+1}$, откуда и получаем нужное условие на l). Более того, найдётся x, для которого все $2n+1$ протоколов имеют длину $\log_{2q} \frac{2^n}{2n+1} - 1 = \Omega(n - \log n) = \Omega(n)$. (Действительно, по принципу Дирихле найдётся i, такой что i-ый протокол будет самым коротким для $\frac{2^n}{2n+1}$ слов, из предыдущего для одного из этих слов он будет иметь длину $\log_{2q} \frac{2^n}{2n+1} - 1$, а все остальные протоколы для этого слова будут не короче). Ну а поскольку каждому символу каждого протокола соответствует отдельный шаг работы машины, общее число шагов на этом x составит $\Omega(n^2)$, что и было заявлено.

Функция называется *вычислимой*, если её вычисляет какая-то машина Тьюринга.

17. Покажите, что множество вычислимых функций не изменится, если запретить машине стоять на месте (т.е. ограничиться машинами с функциями перехода вида $\delta\colon (Q \setminus \{q_0\}) \times \Sigma \to Q \times \Sigma \times \{L, R\}$).

18. Покажите, что множество вычислимых функций не изменится, если ограничить ленту с одной стороны. (Технически можно добавить слева спецсимвол $\rhd$, который нельзя стирать и с которого нельзя двигаться влево).

19. Покажите, что множество вычислимых функций не изменится, если разрешить машине работать на нескольких лентах. (На каждой ленте своя каретка, способная двигаться независимо от других. Формализуйте такую машину самостоятельно).

20. Покажите, что множество вычислимых функций не изменится, если машина работает не на ленте, а на клетчатой плоскости. (Каретка одна, но может двигаться в любом из четырёх направлений).

21. Покажите, что множество вычислимых функций не изменится, если разрешить машине произвольные сдвиги (т.е. функция перехода имеет вид $\delta\colon (Q \setminus \{q_0\}) \times \Sigma \to Q \times \Sigma \times \mathbb{Z}$).

22. Покажите, что множество вычислимых функций не изменится, если разрешить машине произвольный доступ (random access). Формальное определение такое: у машины есть две ленты: адресная и рабочая, специальное состояние q_{access}, а также два спецсимвола `W` и `R` в ленточном алфавите. Если машина оказывается в состоянии q_{access} в конфигурации $\mathrm{bin}(i)q_{access}\mathtt{W}\sigma$ на адресной ленте ($\mathrm{bin}(i)$ — двоичная запись числа i), то символ σ записывается в ячейку с номером i на рабочей ленте. Если машина оказывается в состоянии q_{access} в конфигурации $\mathrm{bin}(i)q_{access}\mathtt{R}$ на адресной ленте, то в следующую за содержащей `R` ячейку записывается символ из ячейки с номером i на рабочей ленте.

23. Пусть машине запрещено стирать символы (кроме #). Как следует определить вычисление функции на такой машине, чтобы множество вычислимых функций не изменилось?

24. Покажите, что множество вычислимых функций уменьшится, если запретить машине сдвигаться влево.

Указание: такая машина не может распознать язык $\{0^n1^n \mid n \in \mathbb{N}\}$. Надо посмотреть на состояние, в которых машина пересекает границу нулей и единиц, для каких-то 0^n1^n и 0^m1^m это будет одно и то же состояние. Далее хотя бы на одном из слов 0^n1^n, 0^n1^m, 0^m1^n и 0^m1^m ответ будет неверным.

25. Покажите, что если разрешить машине иметь бесконечное множество состояний, то любая функция станет вычислимой.

Указание: нужно завести состояние для каждого входного слова, после чтения входа прийти в соответствующее состояние, заодно стирая символы, и вывести правильный ответ, который записан в программе.