# Листок 2

## Классы сложности $\mathcal{R}$ и $\mathcal{RE}$

Детерминированная машина Тьюринга (ДМТ) $M$, останавливающаяся на любом входе, называется детерминированным распознавателем или просто распознавателем. Определение мотивировано тем, что такая ДМТ всегда может быть использована для распознавания некоторого языка: конкретные состояния не столь важны, поскольку могут быть сгруппированы в одно метасостояние «принять» и одно «отвергнуть». Даже, если состояние у ДМТ, останавливающейся на любом входе, всего одно, – это тоже распознаватель: для $\Sigma^*$ либо для $\varnothing$.

Если распознаватель $M$ останавливается в состоянии «принять» на словах $x$ некоторого языка $L$ и только на них (это, в том числе, значит, что на всех словах, не принадлежащих языку $L$, распознаватель **останавливается** не в состоянии «принять» – подобное поведение без ограничения общности можно считать остановом в состоянии «отвергнуть»), язык $L$ называется **разрешимым** или рекурсивным. Множество разрешимых языков $\mathcal{R}$ – в точности класс языков, для которых существуют распознаватели.

Более формально, $L \in \mathcal{R}$ означает, что существует ДМТ $M$ такая, что для любого слова $x \in \Sigma^*$

– если $x \in L$, то $M$ останавливается на $x$ в состоянии «принять»,

– если $x \notin L$, то $M$ останавливается на $x$ в состоянии «отвергнуть».

Далее мы будем считать термины «машина $M$ принимает слово $x$», «машина $M$ останавливается на слове $x$ в состоянии $q_{accept}$» и «$M(x) = 1$» синонимами.

Аналогично синонимами являются «машина $M$ отвергает слово $x$», «машина $M$ останавливается на слове $x$ в состоянии $q_{reject}$» и «$M(x) = 0$».

Напомним также, что язык $L$ называется **рекурсивно перечислимым** или **полуразрешимым** (пишут $L \in \mathcal{RE}$), если для него существует машина Тьюринга, которая остановится и примет любое слово из языка, но остановится и отвергнет или не остановится вообще для любого слова не из языка. Очевидно $\mathcal{R} \subset \mathcal{RE}$. Более того, $\mathcal{R} = \mathcal{RE} \cap \operatorname{co}\mathcal{RE}$.

## Класс сложности $\mathcal{P}$

$L \in \mathcal{P}$ означает следующее: существует ДМТ $M$, существует полином $p(n)$, такие, что

1) для любого $x \in \Sigma^*$ машина $M$ останавливается на входе $x$ не более, чем за $p(|x|)$ шагов

и

2.1) если $x \in L$, то $M(x) = 1$,

2.2) если $x \notin L$, то $M(x) = 0$.

Поскольку машина $M$ останавливается на любом входе, распознавая язык $L$, то в частности $\mathcal{P} \subset \mathcal{R}$.

Распознаватель $M$ называется полиномиальным, если существует некоторый полином $p(n)$, что на каждом слове $x \in \Sigma^*$ $M$ делает не более $p(|x|)$ шагов до останова. В таком случае класс $\mathcal{P}$ — это класс языков, для которых существует полиномиальный распознаватель.

Обратим также внимание, что для принадлежности языка классу $\mathcal{P}$ достаточно одной МТ, полиномиально распознающей этот язык.

В терминах временных классов $\mathcal{P} = \bigcup\limits_{k=0}^{\infty} \mathrm{DTIME}(n^k)$ (константа при О-большом неявно учитывается в определении).

## Недетерминированная машина Тьюринга

Напомним, ДМТ – это тройка $(Q, \Gamma, \delta)$ с выделенным начальным состоянием, пустым символом и т.д. Сигнатура (частичной) функции перехода выглядит так: $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, +1\}$.

Как и в случае конечных автоматов, отличия между детерминированной МТ и недетерминированной МТ заключаются в использовании "многозначной" функции перехода. Недетерминированной машиной Тьюринга (НМТ) называется тройка $(Q, \Gamma, \delta)$ с выделенным начальным состоянием, пустым символом и т.д.

Функция перехода при этом определяется одним из двух эквивалентных подходов:

$\delta \subset (Q \times \Gamma) \times (Q \times \Gamma \times \{-1, +1\})$

или

$\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{-1,+1\}}$

Аналогично можно определить многоленточную недетерминированную машину Тьюринга и прочие модификации.

При каждом шаге вычисления переход осуществляется по нескольким (в том числе нулю) путям, формируя корневое дерево вычислений. Временем работы НМТ при этом будет называться глубина этого корневого дерева. Из-за "разрастания" дерева НМТ за полиномиальное время работы в общем случае делает экпоненциально много "стандартных" вычислений.

НМТ $M$ распознаёт язык $L$ если

– для любого входа $x \in \{0,1\}^*$ $M$ **останавливается на каждой ветви вычислений**;

– если $x \in L$, то **хотя бы одна** из ветвей заканчивается в состоянии $q_{accept}$ – тогда машина принимает слово $x$,

– если $x \notin L$, то **все** ветви заканчиваются в состоянии $q_{reject}$ – тогда машина отвергает слово $x$.

По аналогии с ДМТ существуют теоремы о полиномиальной эквивалентности различных вариаций НМТ. Фиксируйте ваше любимое определение НМТ для дальнейшего рассуждения.

Пусть дана НМТ $M$, пусть $T : \mathbb{N} \to \mathbb{R}$. Скажем, что $M$ работает за время $T(n)$, если для любого $x \in \Sigma^*$ машина $M$ завершает работу **на каждой ветви вычислений** (при любом недетерминированном выборе) не более, чем за $T(|x|)$ шагов. Как и ранее, скажем, что НМТ $M$ работает за время $O(T(n))$, если существует функция $T'(n) = O(T(n))$ такая, что $M$ работает за время $T'(n)$.

Пусть $T : \mathbb{N} \to \mathbb{R}$. Говорят, что язык $L \in \mathrm{NTIME}(T(n))$, если существует недетерминированная машина Тьюринга $M$, которая распознает язык $L$ и работает за время $O(T(n))$.

## Класс сложности $\mathcal{NP}$

$L \in \mathcal{NP}$ означает следующее:

– существует НМТ $M$, существует полином $p(n)$, такие, что $M$ работает за время $p(|x|)$ шагов (напоминаю, останавливается на каждой ветви вычислений и на каждой же ветви делает не более полинома шагов);

– если $x \in L$, то $M$ принимает $x$, если $x \notin L$, то $M$ отвергает $x$ (напоминаю, принимает – существует принимающая ветвь, отвергает – все ветви отвергают).

По аналогии с классом $\mathcal{P}$ также можно сказать, что класс $\mathcal{NP}$ — это класс языков, для которых существует полиномиальный недетерминированный распознаватель.

В терминах временных классов $\mathcal{NP} = \bigcup\limits_{k=0}^{\infty} \mathrm{NTIME}(n^k)$

Для класса $\mathcal{NP}$ удобно ввести альтернативное определение.

$L \in \mathcal{NP}$, если: существует **ДМТ** $M$, существует полином $p(n)$, такие, что

1) для любых $x \in \Sigma^*$, $y \in \Sigma^*$ машина $M$ останавливается на входе $(x, y)$ не более, чем за $p(|x|)$ шагов

и

2.1) если $x \in L$, то существует $y \in \Sigma^*$ такой, что $M$ принимает пару $(x, y)$ (мы будем писать $M(x, y) = 1$),

2.2) если $x \notin L$, то для любого $y \in \Sigma^*$ $M$ отвергает пару $(x, y)$ (мы будем писать $M(x, y) = 0$),

Формально напишем пункт 2) как

$$(x \in L) \Leftrightarrow (\exists y \in \Sigma^*\ M(x, y) = 1)$$

Необходимо отметить асимметрию определения: если $x \in L$, то существует **сертификат** принадлежности слова языку $y$, такой что $M(x, y) = 1$, однако если $x \notin L$, то такого сертификата не существует, т.е. для любого слова $y$ $M(x, y) = 0$.

## Класс сложности co$\mathcal{NP}$

Для любого множества языков $\mathcal{C}$ определим множество языков $co\mathcal{C}$ следующим образом: для любого языка $L$ верно, что $L \in co\mathcal{C}$ тогда и только тогда, когда $\Sigma^* \setminus L \in \mathcal{C}$.

Далее дополнение языка $L$ будем обозначать через $\bar{L} = \Sigma^* \setminus L$.

В соответствии с определением, $L \in \mathrm{co}\mathcal{NP}$ тогда и только тогда, когда $\bar{L} \in \mathcal{NP}$. Раскрывая определение класса $\mathcal{NP}$, получим следующее.

$L \in \mathrm{co}\mathcal{NP}$, если: существует **ДМТ** $M$, существует полином $p(n)$, такие, что

1) для любых $x \in \Sigma^*$, $y \in \Sigma^*$ машина $M$ останавливается на входе $(x, y)$ не более, чем за $p(|x|)$ шагов

и

2) $x \in \bar{L}$ тогда и только тогда, когда существует $y \in \Sigma^*$, такой что $M$ принимает пару $(x, y)$.

Второй пункт можно переписать, поскольку $x \in \bar{L}$ означает, что неверно, что $x \in L$:

$2^*$) $x \in L$ тогда и только тогда, когда для любого $y \in \Sigma^*$ машина $M$ отвергает пару $(x, y)$. Формально напишем

$$(x \in L) \Leftrightarrow (\forall y \in \Sigma^* \to M(x, y) = 0)$$

Отметим ещё раз существенную асимметрию в классах $\mathcal{NP}$ и $\mathrm{co}\mathcal{NP}$: для принадлежности $L \in \mathcal{NP}$ необходимо, чтобы

– хотя бы одна ветвь недетерминированных вычислений/хотя бы один сертификат приводили бы к принимающему состоянию для слова из языка,

– любая ветвь недетерминированных вычислений/любой сертификат непригодны для принимающего состояния, а потому ведут к отвергающему состоянию для слова не из языка.

Подобная же асимметрия существует между классами $\mathcal{RE}$ и $\mathrm{co}\mathcal{RE}$.

Открытыми проблемами теории вычислительной сложности являются (по крайней мере, насколько мне известно) проблемы равенства классов: $\mathcal{P} \stackrel{?}{=} \mathcal{NP}$ и $\mathcal{NP} \stackrel{?}{=} \mathrm{co}\mathcal{NP}$

# Задачи

**Задача 2.0:** Это даже не задача:

1) доказать эквивалентность двух определений класса $\mathcal{NP}$,

2) доказать, что $\mathcal{P} \subset \mathcal{NP} \cap \mathrm{co}\mathcal{NP}$,

3) доказать, что $\mathcal{P} = \mathcal{NP}$ влечёт $\mathcal{NP} = \mathrm{co}\mathcal{NP}$,

4) доказать, что $\mathrm{co}\mathcal{NP} \neq \left(2^{\Sigma^*} \setminus \mathcal{NP}\right)$,

5) доказать, что $\mathcal{NP} \subset \bigcup_{k=0}^{\infty} \mathrm{DTIME}\left(2^{n^k}\right)$ (класс, стоящий справа называется $\mathcal{EXPTIME}$ или просто $\mathcal{EXP}$.

**Задача 2.1:** Простые вопросы:

1) Для принадлежности классу $\mathcal{NP}$ не требуется никакого алгоритма получения сертификата, лишь его существование. Почему в таком случае каждый разрешимый язык $L$ не принадлежит $\mathcal{NP}$, ведь для любого слова $x$ можно предъявить очень простой сертификат: 1, если $x \in L$, или 0, если $x \notin L$?

2) Проблема останова МТ неразрешима — это известно (по крайней мере, из курса ТФСиА). Однако, для слов, принадлежащих соответствующему языку, существует простейший сертификат — для пары (МТ, слово) сертификатом будет число $N$ — число шагов до останова. Для любой пары (МТ, слово), такой, что МТ останавливается на этом слове, сертификат очевидно существует. Для любой пары, не принадлежащей языку, никакой сертификат не подойдёт. Следовательно, проблема останова принадлежит классу $\mathcal{NP}$, а значит разрешима на НМТ, т.е. разрешима. Где ошибка в рассуждениях?

3) Пусть $L \in \mathrm{co}\mathcal{NP}$, тогда $\bar{L} \in \mathcal{NP}$ по определению. Тогда для $\bar{L}$ есть МТ $M$ такая, что $M(x,u) = 1$ для $x \in \bar{L}$ и $M(x,u) = 0$ для $x \notin \bar{L}$. Но тогда можно определить новую МТ $M'$, поменяв состояния $M$ на противоположные. Тогда $M'(x,u) = 1$ для $x \notin \bar{L}$, т.е. для $x \in L$ и наоборот. Тогда $L \in \mathcal{NP}$ и $\mathcal{NP} = \mathrm{co}\mathcal{NP}$. Где ошибка в рассуждениях?

4) Почему в определении $\mathrm{co}\mathcal{NP}$ мы не меняем существование МТ на всеобщность (для любой МТ), существование полинома на время работы МТ на всеобщность (для любого полинома) и существование полинома на ограничение длины сертификата на всеобщность (для любого полинома) по сравнению с определением $\mathcal{NP}$?

**Задача 2.2:** Продолжаем изучать определение.

Всюду под «можно ли» подразумевается следующее: останется ли при такой замене класс языков $\mathcal{NP}$ неизменным.

1) В определении МТ останавливается за $p(|x|)$ шагов. Можно ли сделать зависимость полиномиальной от всего входа и написать $p(|x| + |y|)$ шагов?

2) Можно ли ограничить $y$, ведь за полиномиальное время невозможно прочесть очень длинный сертификат?

**Задача 2.3:**

Доказать что языки принадлежат $\mathcal{P}$

– язык всех несвязных графов без циклов (здесь и далее кодировка графа на ваше усмотрение,

если не сказано обратного),

– язык всех квадратных матриц на целых числах, в которых есть подматрица размера $27 \times 27$, заполненная нулями,

– язык всех графов, в которых существует эйлеров цикл и как минимум пять вершин имеют нечётную степень.

**Задача 2.4:**

Докажите, что классы $\mathcal{P}$, $\mathcal{NP}$ и $\mathrm{co}\mathcal{NP}$ замкнуты относительно операций:

а) объединения,

б) пересечения,

в) конкатенации,

г) итерации (звёздочки Клини),

Дополнительно докажите, что $\mathcal{P}$ замкнут относительно дополнения.

**Задача 2.5:**

Формально определить конфигурацию НМТ, вычисление на НМТ в терминах конфигураций, распознавание языка и полиномиальное время работы НМТ в терминах конфигураций.

**Задача 2.6:**

1) Язык PRIMES — это язык простых чисел, двоичная запись натурального числа $x$ принадлежит языку PRIMES тогда и только тогда, когда $x$ простое число. Доказать, что

а) $\mathrm{PRIMES} \in \mathrm{co}\mathcal{NP}$,

б) $\mathrm{PRIMES} \in \mathcal{NP}$,

в)$^{**!!!}$ $\mathrm{PRIMES} \in \mathcal{P}$.

2) Докажите, что язык

$\mathrm{FACTOR} = \{(n, k) \mid n$ содержит делитель, больший 1, но не превосходящий $k\}$

принадлежит классу $\mathcal{NP} \cap \mathrm{co}\mathcal{NP}$.

**Задача 2.7:**

Доказать что языки принадлежат $\mathcal{NP}$.

1) Язык пар графов $(G, H)$ таких, что графы $G$ и $H$ изоморфны.

2) Язык пар чисел $(n, k)$ таких, что $n$ содержит простой делитель $p$, при этом $p \leqslant k$.

3) Язык пар чисел $(n, k)$ таких, что существует разбиение множества $\{1, \ldots, n\}$ на $k$ подмножеств таких, что ни одно подмножество не содержит таких трёх чисел $x, y, z$, что $x + y = z$.

4) Язык всех графов, содержащих эйлеров путь.

5) Язык всех графов, содержащих гамильтонов путь.

6) Язык всех взвешенных орграфов, в которых нет цикла отрицательной длины.

7) Язык всех составных чисел (записаны в десятичной системе счисления).

8) CNF–SAT.

**Задача 2.8:**

Доказать, что языки принадлежат со$\mathcal{NP}$

1) TAUT – язык, состоящий из описаний булевых тавтологий,

2) Язык, состоящий из пар $(G, k)$ где $G$ – описание графа, такого, что для любых $k$ вершин найдется ребро, соединяющее хотя бы 2 из них.

3) Язык описаний графов, в которых есть клика на 40 вершинах (клика – это подмножество вершин графа, таких, что каждая соединена ребром с каждой).

4) PLANARITY – язык описаний планарных графов.

**Задача 2.9:**

1) Привести пример языка $L$ такого, что он не принадлежит $\mathcal{P}$.

2) Привести пример языка $L$ такого, что он не принадлежит $\mathcal{P}$, но принадлежит $\mathcal{R}$.

3) Привести пример языка $L$ такого, что он не принадлежит $\mathcal{P}$, а $L^*$ принадлежит $\mathcal{P}$.