Летняя школа “Современная математика”
18–29 июля 2008 года

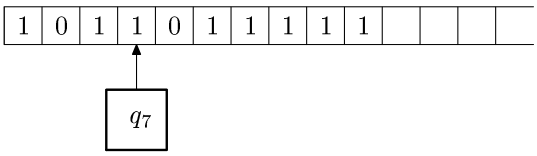
Юрий Притыкин

Что такое проблема P vs NP?

Определения и задачи

Занятие 1. Машины Тьюринга

Машина Тьюринга — это совокупность семи объектов $\langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \rangle$, где Q, Σ, Γ — конечные множества, причём Q — множество (внутренних) состояний, Σ — входной алфавит, не содержащий специального символа пробела $\sqcup$, Γ — ленточный алфавит, содержащий символ $\sqcup$ и алфавит Σ, $\delta\colon Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ — функция переходов, $q_0 \in Q$ — начальное состояние, $q_{\text{accept}} \in Q$ — принимающее состояние, $q_{\text{reject}} \in Q$ — отвергающее состояние, причём $q_{\text{accept}} \neq q_{\text{reject}}$. Машина действует следующим образом: она получает на вход слово $w_1 w_2 \ldots w_n \in \Sigma^*$ записанным слева на бесконечной вправо ленте, правее входного слова лента заполнена символами пробела $\sqcup$. Изначально машина находится в состоянии q_0, головка машины расположена в самой левой клетке ленты. Дальше машина действует по шагам: на каждом следующем шаге если в клетке с головкой машины написан символ $a \in \Gamma$, машина находится в состоянии $q \in Q$, и $\delta(q, a) = (q', b, D)$, то машина записывает в клетку, в которой расположена головка, символ b, меняет внутреннее состояние на q' и перемещает головку на одну клетку влево, если $D = L$, или на одну клетку вправо, если $D = R$. (Попытка сместить головку влево из самой левой клетки машиной игнорируется.) Если внутреннее состояние машины становится q_{accept}, машина завершает работу и принимает изначальное входное слово, если внутреннее состояние становится q_{reject}, машина завершает работу и отвергает слово; иначе машина продолжает работать. Конфигурацией машины Тьюринга называется совокупность внутреннего состояния машины, полного состояния ленты и положения головки машины на ленте. На рисунке приведён пример машины в некоторой конфигурации.



Пусть фиксирован конечный алфавит Σ. Слово — это произвольная конечная последовательность букв алфавита. Язык — это произвольное множество слов. Язык всех слов в алфавите Σ обозначается Σ^*. Язык называется разрешимым, если он разрешается какой-нибудь машиной Тьюринга, то есть если эта машина принимает любое слово этого языка и отвергает любое слово, не принадлежащее языку. Язык называется принимаемым, если он принимается какой-нибудь машиной Тьюринга, то есть если эта машина принимает все слова этого языка и не принимает никаких других (но, возможно, и не отвергает). Многие (любые конечно описываемые) объекты можно кодировать словами в подходящем алфавите: числа, многочлены, графы, формулы, конечные множества, машины Тьюринга, и т. д. Такое кодирование мы обозначаем угловыми скобками $\langle\cdot\rangle$. Таким образом, мы можем рассматривать не только задачи о словах, но и о числах, графах, других объектах.

1. а) Как кодировать словами в двоичном алфавите **б)** пары слов **в)** последовательности произвольной длины слов в двоичном алфавите? Как кодировать словами конечного алфавита **г)** натуральные числа? **д)** графы? **е)** Как кодировать числа и графы двоичными словами?

2. Опишите машину Тьюринга, разрешающую язык $\{w\#w : w \in \{0,1\}^*\}$; **а)** $\{0^{2^n} : n \in \mathbb{N}\}$; **б)** $\{a^i b^j c^k : i + j = k,\ i, j, k \geqslant 0\}$; **в)** $\{a^i b^j c^k : ij = k,\ i, j, k \geqslant 1\}$; **г)** $\{w :$ в слове $w \in \{0,1\}^*$ одинаковое количество 0 и 1$\}$; **д)** $\{w :$ в слове $w \in \{0,1\}^*$ количество 0 в два раза больше, чем количество 1$\}$; **е)** $\{w :$ в слове $w \in \{0,1\}^*$ количество 0 не более чем в два раза больше, чем количество 1$\}$; **ж)** $\{\langle p, q\rangle :$ квадратное уравнение $x^2 + px + q = 0$ имеет корень$\}$; **з)** $\{\langle G\rangle :$ G — связный неориентированный граф$\}$.

3. а) Опишите машину Тьюринга, копирующую входное слово. **б)** Докажите, что для любой такой машины найдётся $\varepsilon > 0$, такое что найдутся сколь угодно длинные слова, на которых машина работает не быстрее, чем за εn^2 шагов, где n — длина слова.

4. Докажите, что **а)** объединение **б)** пересечение **в)** разность **г)** конкатенация **д)** дополнение разрешимых языков является разрешимым языком. **е)** Верно ли всё то же для принимаемых языков?

5. а) Существуют ли неразрешимые языки? **б)** Существуют ли неразрешимые принимаемые языки?

6. Придумайте разные модификации определения машины Тьюринга (количество и вид лент, конечная память, куда и как ходить, взрыв машины при попытке пойти влево из самой левой клетки, и т. п.) и докажите, что все они эквивалентны исходному (в том смысле, что распознаваемые ими языки одни и те же).

7. В недетерминированной машине Тьюринга (в отличие от ранее описанной детерминированной) функция переходов многозначная, и разрешён переход по любому из значений. Слово принимается, если оно принимается хотя бы одним каким-нибудь способом. Докажите, что классы принимаемых языков в детерминированном и недетерминированном случае равны.

Занятие 2. Классы P и NP

Пусть $f, g\colon \mathbb{N} \to \mathbb{N}$ — функции. Говорят, что "f есть о большое от g при n стремящемся к бесконечности" и пишут $f = O(g)$, если существует такая константа $C > 0$ и такое $n_0 \in \mathbb{N}$, что для всех $n > n_0$ выполнено $f(n) \leqslant Cg(n)$.

Мы говорим, что машина Тьюринга работает за полиномиальное время, если для некоторого k максимальное её время работы на входе длины n есть $O(n^k)$ при $n \to \infty$.

Недетерминированная машина Тьюринга называется разрешающей, если она заканчивает работу на каждой своей ветви вычисления. Разрешающая машина Тьюринга работает за полиномиальное время, если для некоторого k максимальное время её работы на любой ветви вычисления на входе длины n есть $O(n^k)$ при $n \to \infty$.

Класс всех языков, для которых существует разрешающая их машина Тьюринга, работающая за полиномиальное время, обозначается P (от слов polynomial time). Класс всех языков, для которых существует разрешающая их недетерминированная машина Тьюринга, обозначается NP (от слов nondeterministic polynomial time). Как несложно видеть, $\mathrm{P} \subseteq \mathrm{NP}$.

8. Докажите, что язык $A = \{0^k 1^k : k \geqslant 0\}$ лежит в классе P. Какова его реальная сложность (асимптотически)?

9. Докажите, что следующие языки лежат в классе P: **а)** $PATH = \{\langle G, s, t\rangle$: в ориентированном графе G есть путь из s в $t\}$; **б)** $RELPRIME = \{\langle x, y\rangle : x$ и y взаимно просты$\}$; **в)** $MODEXP = \{\langle a, b, c, p\rangle : a, b, c, p$ — записанные в двоичной системе счисления натуральные числа, такие что $a^b \equiv c \pmod{p}\}$.

10. Докажите, что следующие языки лежат в классе NP: **а)** $HAMPATH = \{\langle G, s, t\rangle$: в неориентированном графе G есть гамильтонов (проходящий по всем вершинам ровно по одному разу) путь из s в $t\}$; **б)** $COMPOSITES = \{x : x = pq$ для каких-то целых чисел $p, q > 1\}$; **в)** $CLIQUE = \{\langle G, k\rangle :$ G содержит полный подграф с k вершинами$\rangle\}$; **г)** $SUBSETSUM = \{\langle S, t\rangle : S = \{x_1, \ldots, x_k\}$, и для некоторого подмножества $\{y_1, \ldots, y_l\} \subseteq \{x_1, \ldots, x_k\}$ выполнено $\sum y_i = t\}$.

11. Верно ли, что $COMPOSITES \in \mathrm{P}$?

12. Все слова языка L принимаются машиной Тьюринга M за полиномиальное время, и никакие другие слова машиной M не принимаются (но, возможно, и не отвергаются). Верно ли, что L лежит в P?

Проблема: равны ли классы P и NP?

В настоящее время эта проблема — одна из основных задач theoretical computer science и математики вообще. Обычно она датируется 1971 годом (поставлена независимо Левиным и Куком) и с тех пор остаётся открытой. Большинство исследователей верят в то, что $\mathrm{P} \neq \mathrm{NP}$. В этом предположении доказано много теорем, написано много статей и книг.

Занятие 3. Полиномиальные верификаторы. NP-полные задачи

Верификатором для языка A называется такой алгоритм V, что $A = \{w : V$ принимает пару $\langle w, c\rangle$ для некоторого слова $c\}$. При этом слово c называется доказательством или сертификатом принадлежности слова w языку A. Верификатор полиномиальный, если время его работы на входе $\langle w, c\rangle$ есть $O(n^k)$ для некоторого k, где n — длина слова w. Заметим, что длина сертификата для полиномиального верификатора тоже есть полином от слова языка (за полиномиальное время машина может пройти только по полиномиальному количеству клеток ленты).

13. Докажите, что язык лежит в классе NP тогда и только тогда, когда у него есть полиномиальный верификатор.

14. Про все языки из класса NP, встречавшиеся до этого, докажите их принадлежность NP с помощью альтернативного определения из предыдущей задачи.

Символы $\wedge$, $\vee$, $\neg$ обозначают логическое И, ИЛИ и НЕ соответственно (вместо $\neg$ используют также черту сверху $\bar{\ }$). С помощью этих операций из переменных, принимающих значения ИСТИНА и ЛОЖЬ, можно составлять формулы. Например, $(\bar{x} \wedge y) \vee (x \wedge \bar{x})$ или $(z \vee \bar{y}) \wedge (x \vee z \vee \bar{y}) \wedge y$. Формула называется выполнимой, если существует такой набор присвоений переменным значений ИСТИНА и ЛОЖЬ, так что значение формулы становится ИСТИНА. $SAT = \{\langle\phi\rangle : \phi$ — выполнимая формула$\}$, $kSAT = \{\langle\phi\rangle : \phi$ — выполнимая формула, являющаяся логическим И скобок, каждая из которых есть логическое ИЛИ k переменных или их отрицаний$\}$.

15. Докажите, что SAT и $kSAT$ для любого k лежат в NP.

16. Докажите, что $2SAT$ лежит в P.

Функция $f\colon \Sigma^* \to \Sigma^*$ полиномиально вычислима, если существует такая машина Тьюринга, работающая за полиномиальное время, которая, получив на вход слово w, завершает работу со словом $f(w)$ на ленте. Мы говорим, что язык $A \subseteq \Sigma^*$ полиномиально сводится к языку $B \subseteq \Sigma^*$ и пишем $A \leqslant_P B$, если существует такая полиномиально вычислимая функция $f\colon \Sigma^* \to \Sigma^*$, называемая сводимостью, что $w \in A \Leftrightarrow f(w) \in B$.

17. а) Докажите, что если $A \leqslant_P B$ и $B \in$ P, то $A \in$ P. **б)** Докажите, что если $A \leqslant_P B$ и $B \in$ NP, то $A \in$ NP. **в)** Докажите, что если $A \leqslant_P B$ и $B \leqslant_P C$, то $A \leqslant_P C$.

18. Докажите следующие сводимости: **а)** $3SAT \leqslant_P SAT$; **б)** $SAT \leqslant_P 3SAT$; **в)** $3SAT \leqslant_P CLIQUE$; **г)** $3SAT \leqslant_P HAMPATH$.

Язык A называется NP-полным, если $A \in$ NP и для любого $B \in$ NP выполнено $B \leqslant_P A$.

19. Докажите, что если какой-то NP-полный язык лежит в P, то P $=$ NP.

20. а) Докажите, что SAT NP-полный. **б)** Докажите, что $HAMPATH$ NP-полный.

21. Докажите, что если A является NP-полным, язык B лежит в NP и $A \leqslant_P B$, то B является NP-полным.

22. Предположим, P $\neq$ NP. Докажите тогда, что в NP существуют языки, которые не лежат в P, но и не являются NP-полными.

Занятие 4. Приближённые алгоритмы

До сих пор мы подразумевали, что нам необходимо решать задачи точно. Оказывается, иногда гораздо легче решить задачу приближённо.

23. Назовём покрытием неориентированного графа такое подмножество множества его вершин, что каждое ребро графа соприкасается хотя бы с одной вершиной этого подмножества. Пусть $VERTEXCOVER = \{\langle G, k\rangle :$ граф G имеет покрытие размера $k\}$. **а)** Докажите, что $VERTEXCOVER \in$ NP. **б)** Докажите, что язык $VERTEXCOVER$ является NP-полным. Из этого следует, что задача поиска минимального покрытия графа сложная. **в)** Докажите, что можно за полиномиальное время находить покрытие графа, не больше чем в два раза превосходящее минимальное.

24. Назовём разрезом неориентированного графа разбиение множества его вершин на два подмножества, и назовём размером этого разреза количество рёбер, соединяющих вершины из разных подмножеств. $MAXCUT = \{\langle G, k\rangle :$ граф G имеет разрез размером не меньше $k\}$. **а)** Докажите, что язык $MAXCUT$ является NP-полным. **б)** Докажите, что можно за полиномиальное время находить разрез в графе, меньший максимального не более чем в два раз.

Вероятностная машина Тьюринга — это недетерминированная машина Тьюринга, у которой каждый недетерминированный шаг имеет два равновероятных варианта. Такой шаг будем называть бросанием монеты. Вероятностью ветви вычисления мы считаем число 2^{-k}, где k — количество бросаний монеты на этой ветви. Вероятностью $\Pr(M$ принимает $w)$ того, что вероятностная машина M принимает слово w, назовём сумму вероятностей всех ветвей вычисления этой машины, на которых слово w принимается. Разрешающая вероятностная машина M разрешает язык A с вероятностью ошибки ε, если при $w \in A$ выполнено $\Pr(M \text{ принимает } w) \geqslant 1 - \varepsilon$, и при $w \notin A$ выполнено $\Pr(M \text{ принимает } w) \leqslant \varepsilon$. Вероятность ошибки ε может зависеть от длины входа n. Класс всех языков, которые разрешаются за полиномиальное время (на всех входах на всех ветвях вычисления) с вероятностью ошибки $\frac{1}{3}$, обозначим BPP (bounded-error probabilistic polynomial-time).

25. а) Докажите, что если константу $\frac{1}{3}$ в определении класса BPP заменить на любую другую константу строго между 0 и $\frac{1}{2}$, класс не изменится. **б)** Докажите, что если в определении класса BPP взять вероятность ошибки $\varepsilon = 2^{-\mathrm{poly}(n)}$, где $\mathrm{poly}(n)$ — произвольный полином, зависящий от n, то класс не изменится.

В некотором смысле вероятностным аналогом класса NP являются интерактивные доказательства. В схеме интерактивного доказательства есть Доказывающий P (prover) и Проверяющий V (verifier). Вычислительные возможности Проверяющего — полиномиально ограниченная по времени вероятностная машина Тьюринга, вычислительные возможности Доказывающего не ограничены. Им известно входное слово w длины n. Они обмениваются сообщениями полиномиальной от n длины, зависящими от всей истории обмена сообщениями, пока Проверяющий не примет или не отвергнет слово. Он должен сделать это за полиномиальное от n время. Вероятностью $\Pr(V \leftrightarrow P \text{ принимает слово } w)$ называется сумма вероятностей по всем ветвям вычисления на слове w, в которых Проверяющий принял слово. Язык A принадлежит классу IP (interactive proofs), если для него существует схема интерактивного доказательства $V \leftrightarrow P$, такая что при $w \in A$ выполнено $\Pr(V \leftrightarrow P \text{ принимает слово } w) \geqslant \frac{2}{3}$ и при $w \notin A$ выполнено $\Pr(V \leftrightarrow P \text{ принимает слово } w) \leqslant \frac{1}{3}$.

26. а) $ISO = \{\langle G, H\rangle : \text{графы } G \text{ и } H \text{ изоморфны}\}$. Докажите, что $ISO \in \mathrm{NP}$. **б)** $NONISO = \{\langle G, H\rangle : \text{графы } G \text{ и } H \text{ не изоморфны}\}$. Докажите, что $NONISO \in \mathrm{IP}$.

Фиксируем алфавит Σ. Класс coNP состоит из всех языков в алфавите Σ, дополнения к которым до Σ^* лежат в NP. Класс PSPACE (polynomial space) состоит из всех языков, которые разрешимы машиной Тьюринга с полиномиальной памятью. EXPTIME (exponential time) состоит из всех языков, разрешимых за экспоненциальное время, то есть за время $2^{O(n^k)}$ для какого-нибудь k.

27. Докажите включения (это известные, но иногда трудные результаты): **а)** $\mathrm{P} \subseteq \mathrm{NP}$; **б)** $\mathrm{P} \subseteq \mathrm{coNP}$; **в)** $\mathrm{P} \subseteq \mathrm{BPP}$; **г)** $\mathrm{NP} \subseteq \mathrm{IP}$; **д)** $\mathrm{BPP} \subseteq \mathrm{IP}$; **е)** $\mathrm{NP} \subseteq \mathrm{PSPACE}$; **ж)** $\mathrm{BPP} \subseteq \mathrm{PSPACE}$; **з)** $\mathrm{IP} = \mathrm{PSPACE}$; **и)** $\mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$; **к)** $\mathrm{P} \subsetneq \mathrm{EXPTIME}$.

28. Здесь мы перечисляем трудные открытые проблемы теории сложности вычислений. В каждом пункте надо доказать или опровергнуть соотношение (указана более общепринятая гипотеза).
а) $\mathrm{P} \subsetneq \mathrm{NP}$; **б)** $\mathrm{BPP} = \mathrm{P}$; **в)** $\mathrm{NP} \neq \mathrm{coNP}$; **г)** $\mathrm{P} \subsetneq \mathrm{NP} \cap \mathrm{coNP}$; **д)** $\mathrm{NP} \subsetneq \mathrm{PSPACE}$; **е)** $\mathrm{P} \subsetneq \mathrm{PSPACE}$.

Язык $A \subseteq \Sigma^*$ называется NP-трудным (NP-hard), если к нему полиномиально сводится любой язык из класса NP.

29. а) Докажите, что $A = \{\langle p\rangle : p$ — многочлен с целочисленными коэффициентами от переменной x, у которого есть целый корень$\}$ принадлежит P. **б)** Докажите, что $B = \{\langle p\rangle : p$ — многочлен с целочисленными коэффициентами от конечного количества переменных, у которого есть набор целых корней$\}$ является NP-трудной. (Доказано, что B неразрешим — это 10-я проблема Гильберта.)