Курс лекций по теоретической криптографии

Тема 3. **Элементы теории сложности вычислений**

Шокуров А.В.

Зачем нам теория сложности?

Стойкость

- теоретико-информационная — против вычислительно неограниченного противника — очень сильная и труднореализуемая
- теоретико-сложностная — против «эффективного алгоритма»[1] — в большинстве случаев вполне достаточна

То есть нас прежде всего будет интересовать вопрос, является ли задача взлома криптографического протокола непосильной для противника, представленного любым эффективным алгоритмом. Теория сложности вычислений (ТСВ) занимается оценкой вычислительной «сложности» математической задачи в терминах трудоёмкости **всех** решающих её алгоритмов.

Если построение и вычисление трудоёмкости конкретного алгоритма, решающего задачу, даёт верхнюю оценку её сложности, то ТСВ пытается получить для неё нижние оценки.

[1] Это понятие определяется рядом тезисов, см. ниже.

Вычислительная задача

(Массовая) задача кодируется множеством строк (пар строк) в некотором конечном алфавите Σ ($|\Sigma| \geqslant 2$).
Далее будем рассматривать только $\Sigma = \{0, 1\} = \mathbb{B}$

Σ^* — множество всех слов в алфавите Σ, то есть $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$
Язык — некоторое множество слов, то есть подмножество в Σ^*

- **Задача распознавания** языка *L* / decision problem
  Массовая задача $\rightsquigarrow$ язык $L \subseteq \mathbb{B}^*$
  Дано: $w \in \mathbb{B}^*$
  Надо: определить $w \stackrel{?}{\in} L$
  Ответ: да / нет (1 / 0)
- **Задача поиска** по отношению *R* / search problem
  Массовая задача
  $\rightsquigarrow$ двуместное отношение $R \subseteq \mathbb{B}^* \times \mathbb{B}^*$
  Дано: $w \in \mathbb{B}^*$
  Надо: найти a : $(w, a) \in R$
  Ответ: $a \in \mathbb{B}^*$ / нет решения
- Задача подсчёта (числа решений) / counting problem
- Задача оптимизации / optimization problem
- …

? Как представить задачу распознавания в виде (частного случая) задачи поиска?

Пример вычислительной задачи

ВЫПОЛНИМОСТЬ / SATISFIABILITY (SAT) — задача о выполнимости булевой формулы

$\Phi(x_1, \ldots, x_n)$ — булева формула от n переменных (в базисе $\wedge, \vee, \neg$)

$\Phi(x_1, \ldots, x_n) \leftrightarrow w_\Phi$ — двоичная строка (слово), кодирующая формулу Φ

- Задача распознавания:
  Выполнима ли данная формула?
  То есть: существует ли такой набор $a = (a_1, \ldots, a_n) \in \mathbb{B}^n$ значений переменных, что $\Phi(a) = \Phi(a_1, \ldots, a_n) = 1$ (a — выполняющий набор для Φ)?

$$L_{\text{SAT}} = \{w_\Phi \mid \exists a\, \Phi(a) = 1\}$$

- Задача поиска:
  Для данной формулы найти выполняющий набор.
  То есть: найти такой набор $a \in \mathbb{B}^n$ значений переменных, что $\Phi(a) = 1$.

$$R_{\text{SAT}} = \{(w_\Phi, a) \mid \Phi(a) = 1\}$$

NB «Сложность» задачи существенно зависит от кодировки:
ср. задание булевой функции таблицей истинности vs. КНФ для задачи SAT.

Модель вычислений

Тезис Тьюринга—Чёрча

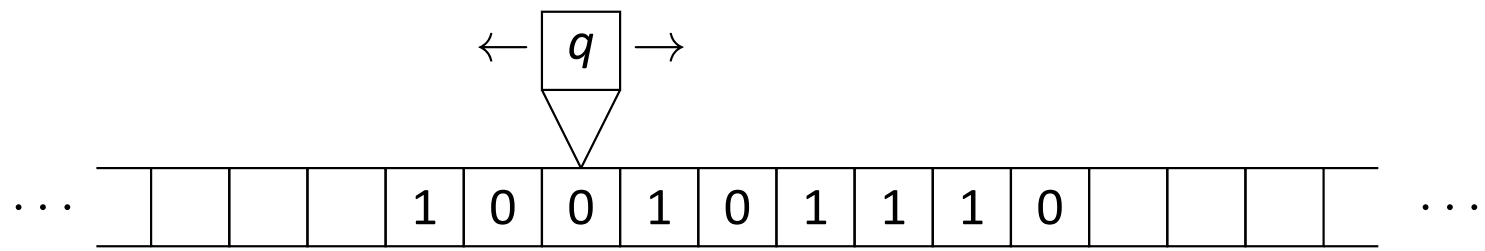
Алгоритм $=$ машина Тьюринга

Модель вычислений, которую мы будем использовать в дальнейшем, — машина Тьюринга

$$M = (Q, q_0, q_f, \Sigma, b, \delta),$$

где

- Q — множество состояний (конечное, непустое),
- $q_0, q_f \in Q$ — выделенные состояния, начальное и конечное (терминальное),
- Σ — конечный алфавит,
- b — специальный «пустой символ»,
- $\delta : \Sigma^k \times Q \to \Sigma^k \times Q \times \{\leftarrow, 0, \rightarrow\}^k$ — функция перехода (частично определённая, в общем случае многозначная).



Решение вычислительной задачи

С машиной Тьюринга M связаны отображения:

- вычисляемая машиной функция $M(\cdot) : \mathbb{B}^* \to \mathbb{B}^* \cup \{\perp\}$,
  $M(w)$ — выход машины M, если на вход подана строка w;
- время её работы $T_M(\cdot) : \mathbb{B}^* \to \mathbb{N} \cup \{\infty\}$,
  $T_M(w)$ — число тактов работы машины M при вычислении на входе w;
- используемая ею память $S_M(\cdot) : \mathbb{B}^* \to \mathbb{N} \cup \{\infty\}$,
  $S_M(w)$ — число ячеек ленты, задействованных в вычислении на входе w.

M решает задачу распознавания языка L, если

$$\forall w \in \mathbb{B}^* \quad M(w) \neq \perp \ \wedge \ (M(w) = 1 \Leftrightarrow w \in L).$$

M решает задачу поиска по отношению R, если

$$\forall w \in \mathbb{B}^* \quad M(w) \neq \perp \ \wedge \ \big(\exists\, a \in \mathbb{B}^* \ (w, a) \in R \Rightarrow (w, M(w)) \in R\big).$$

Мера сложности вычислительных задач

Сложность задачи определяется помещением её в тот или иной «сложностной класс» в зависимости от типа решающей её машины Тьюринга

Тезис Эдмондса

Эффективные (efficient) вычисления — те, которые выполняются за полиномиальное время.

Для произвольной функции $f : \mathbb{N} \to \mathbb{N}$ определяется сложностной класс
$\mathrm{DTIME}(f) = \{L \subseteq \mathbb{B}^* \mid \exists \text{м.т. } M \; \forall w \in \mathbb{B}^* \;\; T_M(w) \leqslant f(|w|) \;\wedge\; (M(w) = 1 \Leftrightarrow w \in L)\}$

$$\begin{array}{ccc} \mathrm{P} & = & \mathrm{PTIME} \\ \rotatebox{90}{$\nparallel$} & & \\ \mathrm{EXP} & = & \mathrm{EXPTIME} \end{array} = \begin{array}{c} \bigcup_{k=1}^{\infty} \mathrm{DTIME}(n^k) \\ \\ \bigcup_{d>0} \mathrm{DTIME}(2^{n^d}) \end{array}$$

P — класс всех эффективно распознаваемых языков

Машины Тьюринга

- *Детерминированная машина Тьюринга*: функция перехода δ однозначна
- *Полиномиальная (детерминированная) машина Тьюринга* M:
$$\exists c,d \in \mathbb{N}\ \forall w \in \mathbb{B}^* \quad T_M(w) \leqslant c \cdot |w|^d$$
$$(\forall w \in \mathbb{B}^* \quad T_M(w) \leqslant \mathrm{poly}\,|w|)^{2}$$

- *Недетерминированная машина Тьюринга*: функция перехода δ вообще говоря многозначна, выбор её значений в конкретном вычислении осуществляется с помощью строки «недетерминированного выбора» $\psi \in \mathbb{B}^\infty$, записанной на специальную ленту
- *Полиномиальная недетерминированная машина Тьюринга* M:
$$\forall w \in \mathbb{B}^*\ \forall \psi \in \mathbb{B}^\infty \quad T_M(\psi; w) \leqslant \mathrm{poly}\,|w|$$

---

[2] $\mathrm{poly}\cdot$ — обозначение для «некоторого полинома» (важен не сам полином, а факт его существования). Нам будет вполне достаточно рассматривать только полиномы вида cn^d.

Машины Тьюринга

- *Вероятностная машина Тьюринга*: функция перехода δ принимает случайные значения, $M(w)$ — случайная величина (при фиксированном w). Выбор значения функции перехода в каждом такте осуществляется с помощью случайной строки $\rho \in_R \mathbb{B}^\infty$, записанной на специальную ленту ($\sim$ подкидывание монеты)
- *Полиномиальная вероятностная машина Тьюринга* (п. в. м. Т.) M:

$$\forall w \in \mathbb{B}^* \; \forall \rho \in \mathbb{B}^\infty \quad T_M(\rho; w) \leqslant \mathrm{poly}\,|w|$$

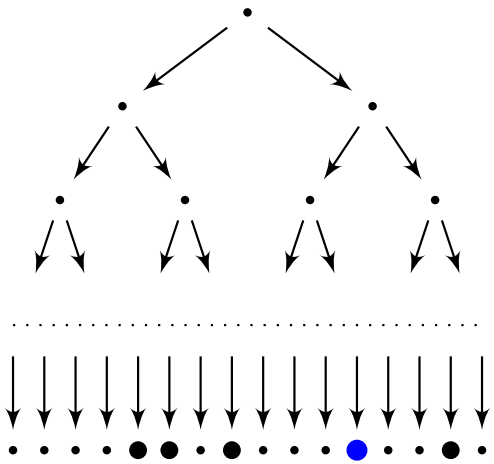
NB Наиболее адекватным считается рассматривать в качестве модели вычислителя противника и «рядовых» честных участников криптографических протоколов именно полиномиальные вероятностные алгоритмы.

Машины Тьюринга

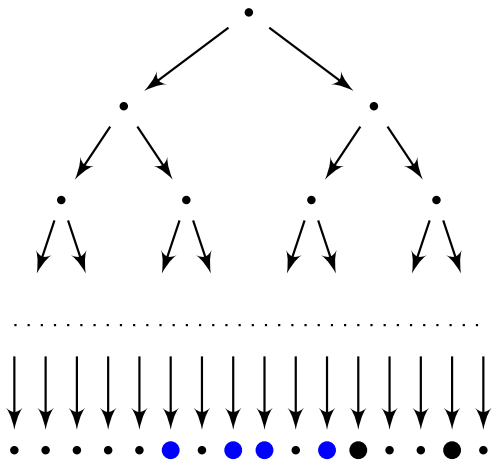
ВЫЧИСЛЕНИЯ

детерминированные

недетерминированные



вероятностные



Машины Тьюринга с оракулом имеют дополнительную ленту, на которую они записывают запрос, а оракул записывает ответ на него (если запрос удовлетворяет всем требованиям) — за один такт!

Классы сложностей

Nondeterministic **P**olynomial time

$$\mathrm{NP} = \left\{ L \subseteq \mathbb{B}^* \mid \exists R \in \mathrm{P}\ \exists \text{полином } p\ \forall w \in \mathbb{B}^* \quad w \in L \Leftrightarrow (\exists a \in \mathbb{B}^{p(|w|)}\ (w, a) \in R) \right\}$$

- класс языков, распознаваемых полиномиальными недетерминированными м. Т.
- класс задач, решаемых переборными методами

Bounded-error **P**robabilistic **P**olynomial time

$$\mathrm{BPP} = \left\{ L \subseteq \mathbb{B}^* \mid \exists \text{ п. в. м. Т. } M \quad \begin{array}{ll} w \in L \Rightarrow & \Pr\left[M(w) = 1\right] \geqslant \frac{2}{3}, \\ w \notin L \Rightarrow & \Pr\left[M(w) = 1\right] \leqslant \frac{1}{3} \end{array} \right\}$$

Randomized **P**olynomial time

$$\mathrm{RP} = \left\{ L \subseteq \mathbb{B}^* \mid \exists \text{ п. в. м. Т. } M \quad \begin{array}{ll} w \in L \Rightarrow & \Pr\left[M(w) = 1\right] \geqslant \frac{2}{3}, \\ w \notin L \Rightarrow & \Pr\left[M(w) = 1\right] = 0 \end{array} \right\}$$

? Покажите, что верна диаграмма

$$\begin{array}{ccccc} \mathrm{P} & \subseteq & \mathrm{RP} & \subseteq & \mathrm{BPP} \\ & & \cap\mathrel{|} & & \cap\mathrel{|} \\ & & \mathrm{NP} & \subseteq & \mathrm{EXP} \end{array}$$

$$\mathrm{SPACE}(f) =$$
$$= \{ L \subseteq \mathbb{B}^* \mid \exists c > 0\ \exists \text{ м.Т. } M\ \forall w \in \mathbb{B}^*\ \ S_M(w) \leqslant c \cdot f(|w|)\ \wedge\ (M(w) = 1 \Leftrightarrow w \in L) \}$$

$$\mathrm{PSPACE} = \bigcup_{k=1}^{\infty} \mathrm{SPACE}(n^k)$$

Модели вычислений

Однородная модель (машина Тьюринга $\rightsquigarrow$ один, но «масштабируемый» алгоритм)

Модель вычислителя противника — эффективный алгоритм — полиномиальная вероятностная машина Тьюринга или полиномиальная *в среднем* вероятностная машина Тьюринга.

(*Полиномиальная в среднем* вероятностная машина Тьюринга:
$\exists \varepsilon > 0 \, \forall n \in \mathbb{N} \, \forall \rho \in \mathbb{B}^\infty \, \forall w \in \mathbb{B}^n \quad \mathsf{E}\left((T_M(\rho; w)^\varepsilon\right) \leqslant n$)

Неоднородная модель (семейство схем $\rightsquigarrow$ своя схема для каждого размера входа)

Модель вычислителя противника — эффективный алгоритм — семейство $C = \{C_n\}$ булевых схем полиномиального размера (сложности): $\exists$ полином $p \, \forall n \quad |C_n| \leqslant p(n)$

Теорема

Язык L распознаётся детерминированной машиной Тьюринга за время $T(n) \Longrightarrow$ существует семейство схем размера $O(T^2(n))$, такое, что

$$\forall w \in \mathbb{B}^n \quad w \in L \Leftrightarrow C_n(w) = 1.$$

$$\mathrm{SIZE}(f) =$$
$$= \{L \subseteq \mathbb{B}^* \mid \exists \{C_n\}_{n \in \mathbb{N}} : \ \forall n \in \mathbb{N} \ |C_n| \leqslant f(n) \ \wedge \ \forall w \in \mathbb{B}^* \ (w \in L \Leftrightarrow C_{|w|}(w) = 1)\}$$

Связь однородной и неоднородной модели

Пусть $A = (a_1, a_2, \ldots)$ — бесконечная последовательность «подсказок» полиномиальной длины, $a_i \in \mathbb{B}^{\leqslant \mathrm{poly}\, i}$.

$\mathrm{P/poly} = \{L \subseteq \mathbb{B}^* \mid \exists R \in \mathrm{P}\ \forall n \in \mathbb{N}\ \exists a_n \in A\ \forall w \in \mathbb{B}^n \quad w \in L \Leftrightarrow (w, a_n) \in R\}$ [3]

Теорема

Язык $L \in \mathrm{P/poly}$ тогда и только тогда, когда L распознаётся семейством схем полиномиального размера, то есть

$$\mathrm{P/poly} = \bigcup_{k=1}^{\infty} \mathrm{SIZE}(n^k).$$

Теорема демонстрирует в некотором смысле эквивалентность двух определений эффективного алгоритма.

Теорема

$$\mathrm{BPP} \subseteq \mathrm{P/poly}$$

[3] В отличие от определения класса NP, здесь подсказки одни и те же для слов одинаковой длины и они подаются на вход алгоритму.

Сводимость задач, их трудность и полнота в классе

Функция $f : \mathbb{B}^* \to \mathbb{B}^*$ называется *полиномиально вычислимой*, если существуют (детерминированная) машина Тьюринга M и полином p, такие, что

$$\forall w \in \mathbb{B}^* \quad T_M(w) \leqslant p(|w|) \ \wedge \ M(w) = f(w).$$

Определение

Язык L' *сводится (по Карпу) за полиномиальное время к языку* L, $L' \leqslant_m^p L$, если существует такая полиномиально вычислимая функция $f : \mathbb{B}^* \to \mathbb{B}^*$, что

$$\forall w \in \mathbb{B}^* \quad w \in L' \Leftrightarrow f(w) \in L.$$

Задача, соответствующая L', — это частный случай задачи, соответствующей L.
Если есть сводимость и в обратную сторону тоже, то это эквивалентные языки и они представляют одну математическую задачу в разных кодировках.

Определение

Если к задаче сводятся все задачи из некоторого сложностного класса K, то эта задача называется K-*трудной*.
Если K-трудная задача сама принадлежит к классу K, то она называется *полной* в этом классе (K-полной).

Теорема (Кук—Левин)

Задача SAT полна в классе NP.

Литература

- *Н. П. Варновский*. Курс лекций по математической криптографии: (Предварительная версия). — М., 2009. — URL: http://cryptography.ru/wp-content/uploads/2014/11/varn_lectures_lor Глава 2.
- *Н. Н. Писарук*. Сложность вычислений и криптография: (Предварительная версия). — Минск, 1999. — URL: http://pisaruk.narod.ru/books/complexity.pdf Часть 1. — Сс. 1–118.
- *А. А. Татузов*. Введение в теорию сложности вычислений: Материалы к курсу. — М., 2017. — URL: http://cryptography.ru/wp-content/uploads/2017/01/ct-notes3.pdf
- *М. Гэри, Д. Джонсон*. Вычислительные машины и труднорешаемые задачи. — Мир, 1982.
- *C. Papadimitriou*. Computational complexity. — Addison-Wesley, 1994.
- *S. Arora, B. Barak*. Computational complexity: A modern approach. — Cambridge University Press, 2009.
- https://complexityzoo.uwaterloo.ca/Complexity_Zoo https://complexityzoo.net/Complexity_Zoo

Машины Тьюринга с несколькими лентами

Определение

Машина Тьюринга — это набор $T = \langle k, \Sigma, \Gamma, \alpha, \beta, \gamma \rangle$, где

- $k \geqslant 1$ — число лент;
- Σ — алфавит лент, $\star \in \Sigma$ — символ-пробел;
- Γ — конечное множество состояний, $S, Q \in \Gamma$ — выделенные состояния: запуск машины и завершение работы;
- α, β, γ — произвольные отображения:

$$\alpha : \Gamma \times \Sigma^k \to \Gamma,$$
$$\beta : \Gamma \times \Sigma^k \to \Sigma^k,$$
$$\gamma : \Gamma \times \Sigma^k \to \{-1, 0, 1\}^k.$$

т. е. отображение α задает новое состояние, отображение β — символы для записи на ленты, γ — перемещение головок.
Таким образом, машина Тьюринга задается таблицей команд размером $|\Sigma|^k \times |\Gamma|$, задающей правила работы машины в соответствии с функциями α, β, γ.