Алгоритмы и модели вычислений.
Задание 4: Сложность вычислений, классы P, NP и co-NP

Сергей Володин, 272 гр.

задано 2014.03.06

Задача 1

1. Докажем, что SAT $\leqslant_m^p$ 3-SAT. SAT $\subset \Sigma^* \ni w = \bigwedge\limits_{i=1}^{k}\left(\bigvee\limits_{j=1}^{l_i}(x_j^i)^{\sigma_j^i}\right)$. Считаем, что w — запись формулы. Построим по данной формуле эквивалентную $u \in$ 3-SAT. Последовательно пройдем по элементам внешней конъюнкции и заменим каждый на эквивалентный (эквивалентные) в смысле выполнимости, содержащие по 3 элемента в дизъюнкции. Обозначаем $y_j^i = (x_j^i)^{\sigma_j^i}$ — отрицание переменной, либо переменная.

   (a) Пусть $l_i < 3$, в скобке $\Phi_0 = y_1^i \vee ... \vee y_{l_i}^i$. Повторим какой-либо (для определенности, первый) y_1^i до трех элементов в скобке. Полченную формулу обозначим за Φ_1. Очевидно, полученная функция будет тождественно равна исходной, так как $a \vee a \equiv a$.

   (b) Пусть $l_i > 3$, в скобке $\Phi_0 = y_1 \vee ... \vee y_l$ (для краткости не пишем индексы по i, они одни и те же для скобки). Заменим это на $\Phi_1 = (y_1 \vee y_2 \vee z_1) \wedge \underbrace{(\overline{z_1} \vee y_3 \vee z_2) \wedge ... \wedge (\overline{z_{l-4}} \vee y_{l-2} \vee z_{l-3})}_{\varphi} \wedge (\overline{z_{l-3}} \vee y_{l-1} \vee y_l)$. Элементы φ в фигурной скобке строятся следующим образом: в середине n-й скобки стоит y_{i+2}, первый элемент — $\overline{z_i}$, последний — z_{i+1}, т.е. $\varphi = \bigwedge\limits_{n=1}^{l-4}(\overline{z_i} \vee y_{i+1} \vee z_{i+1})$. Докажем, что

$$\forall\{x_j\}_{j=1}^m \hookrightarrow \big(\Phi_0(\{x_j\}) = 1 \Leftrightarrow \exists\{z_j\}\colon \Phi_1(\{x_j\},\{z_j\}) = 1\big)$$

      i. Пусть $\Phi_0(\{x_j\}_{j=0}^m) = 1$. Поскольку Φ_0 — дизъюнкция элементов y_j, то $\exists j\colon y_j = 1$

         A. Если $j \in \{1, 2\}$, то определим все $z_j = 0$. Первая скобка содержит y_j, поэтому истинна. В остальных скобках есть отрицание $\overline{z_j}$, поэтому они тоже истинны.

         B. $j \in l-1, l$. Определим все $z_j = 1$. Последняя скобка истинна, так как содержит y_j, все предыдущие содержат некоторый z_j, поэтому истинны.

         C. Оставшиеся случаи (y_j в формуле φ). Скобка с этим y_j: $\overline{z_{j-2}} \vee y_j \vee z_{j-1}$. Определим слева ($w \leqslant j-2$) все $z_w = 1$, справа ($w \geqslant j-1$) $z_w = 0$. Рассматриваемая скобка истинна, так как содержит y_j, скобки слева истинны, так как содержат $z_w = 1$, скобки справа истинны, так как содержат $\overline{z_w} = \overline{0} = 1$.

      ii. (контрапозиция) Пусть $\Phi_0(\{x_j\}) = 0$. Поскольку эта формула — дизъюнкция y_j, то $y_j = 0$, $j \in \overline{1,l}$. Предположим истинность противоположное доказываемому утверждения, т.е. $\exists\{z_j\}\colon \Phi_1(\{x_j\},\{z_j\}) = 1$. Перепишем формулу с учетом $y_j = 0$: $\Phi_1 = z_1 \wedge \underbrace{(\overline{z_1} \vee z_2) \wedge ... \wedge (\overline{z_{l-4}} \vee z_{l-3})}_{\varphi} \wedge \overline{z_{l-3}}$. Значение равно 1, поэтому все конъюнкты истинны, получаем $z_1 = 1$. Но вторая скобка также истинна, поэтому $z_2 = 1$. Продолжая (по индукции) получаем, что все $z_j = 1$. Но тогда последний конъюнкт $\overline{z_{l-3}} = 0$, и значение формулы — 0 — противоречие.

   (c) Если $l_i = 3$, $\Phi_1 \stackrel{\text{def}}{=} \Phi_0$.

   Определим $u = \Phi_1^i$. $u = \Phi_1^1 \vee ... \vee \Phi_1^k$ — конъюнкция всех полученных Φ_1^i. Тогда для u выполнено то же свойство, что и для каждого Φ_1:

$$\forall\{x_j\}_{j=1}^m \hookrightarrow \big(w(\{x_j\}) = 1 \Leftrightarrow \exists\{z_j\}\colon u(\{x_j\},\{z_j\}) = 1\big).$$

   Действительно:

   (a) Пусть $w(\{x_j\}) = 1$. Тогда выберем $\{z_j\}$ для каждой из формул Φ_1^i в соответствии с алгоритмом выше. Получим, что все $\Phi_1^i = 1$ на полученном наборе.

   (b) Пусть $\exists\{z_j\}\colon u(\{x_j\},\{z_j\}) = 1$. Внешняя операция в u — конъюнкция, поэтому все $\Phi_1^i = 1$. Тогда $\Phi_0^i(\{x_j\}) = 1$ (утверждение для отдельных Φ_1). Но w — конъюнкция Φ_0^i, поэтому $w(\{x_j\}) = 1$.

   Заметим, что из доказанного свойства следует утверждение: w — выполнима $\Leftrightarrow$ u — выполнима.

   (a) Оценим длину получившейся формулы. Каждый из элементов y_j добавляет не более одной скобки (ее длина не превышает константы c_1). С другой стороны, для исходной формулы $|u| \geqslant c_2 \times n$ (каждый y_j имеет ненулевую длину записи), где n — общее количество y_j. Поэтому $|w| \leqslant |u| + c_1 n \leqslant |u| + |u|\frac{c_1}{c_2} = O(|u|) = \mathrm{poly}(u)$.

(b) Определим $f\colon \mathrm{SAT} \subset \Sigma^* \to \Sigma^* \supset 3\text{-}\mathrm{SAT}\colon f(w) = u$ (процедура построения u описана выше). Тогда f вычислима за полиномиальное время. Действительно, алгоритм состоит из k шагов (количество скобок), на каждом шаге скобки модифицируются. Добавляется не более k скобок (в каждой новой скобке есть уникальный y_j из старой скобки). Добавление новой скобки занимает не более, чем $O(|u|)$ (записать строку длины $\leqslant |u|$). Поэтому $T(f(w)) = O(k^2|u|)$. Но $|u| = \Omega(k)$, так как каждая скобка имеет непустую запись в исходной формуле, откуда $T(f(w)) = O(|u|^3)$

(c) Определим $f(w) =$ (, если w — не запись формулы. Тогда $f(w)$ — также не запись формулы (можно проверить за poly).

(d) Итак, построена полиномиально-вычислимая функция $f\colon \Sigma^* \to \Sigma^*$, причем $u \in \mathrm{SAT} \Leftrightarrow f(u) \in 3\text{-}\mathrm{SAT}$ ■

2. Теорема $\Rightarrow \mathrm{SAT} \in \mathsf{NP}\text{-c}$, $1 \Rightarrow \mathrm{SAT} \leqslant_m^p 3\text{-}\mathrm{SAT} \in \mathsf{NP}$. Поэтому из 2 следует, что $3\text{-}\mathrm{SAT} \in \mathsf{NP}\text{-c}$

Задача 2

Пусть $w \in \Sigma^* \supset 2\text{-}\mathrm{SAT}$. Если w — не запись формулы в нужном виде (можно проверить за полиномиальное время), останавливаем МТ в не принимающем состоянии. Далее считаем, что w — запись формулы: $w = (a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge \ldots \wedge (a_n \vee b_n)$, где a_i, b_i — переменная x_j, либо ее отрицание $\overline{x_j}$. Заметим, что $(a \vee b) \equiv (\overline{a} \Rightarrow b) \wedge (\overline{b} \Rightarrow a)$. Построим граф с вершинами $V = \{x_j\}_{j=1}^m \cup \{\overline{x_j}\}_{j=1}^m$. Есть ребро $(\overline{a}, b) \in E \Leftrightarrow$ эквивалентная запись в одной из скобок содержит $\overline{a} \Rightarrow b$. В полученном графе могут существовать пути из x_i в $\overline{x_i}$ и обратно. Докажем утверждение: w — невыполнима $\Leftrightarrow \exists i\colon x_i \to^* \overline{x_i}, \overline{x_i} \to^* x_i$ ($a \to^* b$ — есть путь из a в b)

1. $\boxed{\Leftarrow}$ (контрапозиция). Пусть формула выполнима (на наборе $\{x_i\}$), но $\exists i\colon x_i \to^* \overline{x_i}, \overline{x_i} \to^* x_i$. Пусть $x_i = 1$. Тогда все скобки равны 1, в том числе и те, которые содержат $\overline{x_i}$. Они эквивалентны $(x_i \Rightarrow \cdot) \wedge (\bar{\cdot} \Rightarrow \overline{x_i})$ (см. выше). Обе скобки истинны, поэтому $\cdot = 1$. Но это соответствует ребру в графе. Повторяя рассуждение, получаем, что все y_j, соответствующие вершинам, достижимым из x_i, истинны, в том числе и $\overline{x_i}$ — противоречие. Аналогично получаем противоречие в случае $x_i = 0$ ■

2. $\boxed{\Rightarrow}$ (контрапозиция) $\forall i \in \overline{1, m} \hookrightarrow x_i \not\to^* \overline{x_i}$ или $\overline{x_i} \not\to^* x_i$. Определим в первом случае $x_i = 1$, а во втором определим $x_i = 0$. Выполним поиск в глубину из x_i или $\overline{x_i}$ соответственно. Устанавливаем значения вершин в 1. В случае конфликта (установлены $x_k = 1$, $\overline{x_k} = 1$) отбрасываем найденный путь и «забываем» установленные значения
   После всех таких обходов будет найдей хотя бы один набор значений входящих в дерево переменных, так как в противном случае (возникают конфликты на каждом пути из каждой вершины) получим $x_i \to^* \overline{x_i}, \overline{x_i} \to^* x_i$ (поиск запускается из каждой вершины, поэтому каждое ребро может быть пройдено в обе стороны). Тогда на данном наборе формула истинна, так как истинны все следствия в эквивалентных скобках $(\overline{a} \Rightarrow b) \wedge (\overline{b} \Rightarrow a)$

Алгоритм: строим граф, поиском в глубину ищем пути $a \to^* \overline{a}$. Если найдено $a \to^* \overline{a}$ и $\overline{a} \to^* a$, $w \notin 2\text{-}\mathrm{SAT}$, иначе $w \in 2\text{-}\mathrm{SAT}$. Оценим время работы. Длина входа $|w| = \Omega(n)$. Количество вершин не более $2n$, количество ребер не больше $4n^2$. Количество поисков — $2n$. Тогда $T(w) = O(2n(|V| + |E|)) = O(2n \times 2n + 2n \times 4n^2) = O(n^3) = O(|w|^3)$.

(каноническое) Задача 16

1. $$\left\|A \quad b\right\|^{\square} = \begin{Vmatrix} \underline{1/1} & 0/1 & 0/1 & 4/1 \\ 3/1 & 4/1 & 0/1 & 16/1 \\ 9/1 & 3/1 & 1/1 & 64/1 \end{Vmatrix} \overset{r1*=\frac{1}{1}}{\sim} \begin{Vmatrix} 1/1 & 0/1 & 0/1 & 4/1 \\ 3/1 & 4/1 & 0/1 & 16/1 \\ 9/1 & 3/1 & 1/1 & 64/1 \end{Vmatrix} \underset{\substack{\frac{16}{1} - \frac{4}{1}\frac{3}{1} = \frac{4}{1} \\ \frac{64}{1} - \frac{4}{1}\frac{9}{1} = \frac{28}{1}}}{\overset{s12\frac{3}{1}, s13\frac{9}{1}}{\sim}} \begin{Vmatrix} 1/1 & 0/1 & 0/1 & 4/1 \\ 0/1 & \underline{4/1} & 0/1 & 4/1 \\ 0/1 & 3/1 & 1/1 & 28/1 \end{Vmatrix} \overset{r2*=\frac{1}{4}}{\sim}$$
$$\sim \begin{Vmatrix} 1/1 & 0/1 & 0/1 & 4/1 \\ 0/1 & 1/1 & 0/1 & 1/1 \\ 0/1 & 3/1 & 1/1 & 28/1 \end{Vmatrix} \underset{\frac{28}{1} - \frac{1}{1}\frac{3}{1} = \frac{25}{1}}{\overset{s23\frac{3}{1}}{\sim}} \begin{Vmatrix} 1/1 & 0/1 & 0/1 & 4/1 \\ 0/1 & 1/1 & 0/1 & 1/1 \\ 0/1 & 0/1 & 1/1 & 25/1 \end{Vmatrix}$$
   (a) $r_i* = \frac{c_1}{c_2}$ — умножение строки i на дробь $\frac{c_1}{c_2}$
   (b) $sij\frac{c_1}{c_2}$ — вычитание i-й строки, умноженной на дробь $\frac{c_1}{c_2}$ из j-й.

2. $d_{33}^{(2)} = 1 \times 1 \times 1 = 1 = 1 \times 1 \times 1 = d_1 d_2 a_{33}^{(2)}$

(каноническое) Задача 17

1. (не дописано) Сертификат — решение системы уравнений. Длина полиномиальна (???). Проверочный сертификат: подставляем числа, проверяем равенства. Время проверки полиномиально (???)

2. Неравенства сводятся к равенствам путем увеличения количества переменных. Действительно, $a_1x_1 + \ldots + a_nx_n \geqslant b_n \Leftrightarrow a_1x_1 + \ldots + a_nx_n = b_n + x$, где $x \geqslant 0$. Дополнительно в сертификат входит список переменных x, для которых должно быть выполнено $x > 0$.

3. Сертификат для x — двоичная запись делителя y. $R(x, y) =$ „$y|x$” $\vee$ „$y \geqslant 2$” $\vee$ „в x не встречается 10101”. $y < x$, поэтому $|y| \leqslant |x|$. Первая часть проверяется за $O(|x|^2)$ (деление в столбик), вторая часть — за $O(1)$, третья — за $O(|x|)$ — поиск подстроки.

(каноническое) Задача 18

1. Сертификат простоты 3361. Первообразный корень $g = 22$. Делители $p-1 = 2^3 \cdot 3 \cdot 5 \cdot 7$ — $\underline{2, 3, 5, 7}$. Далее сертификаты для простых делителей $p-1$

   (a) $2, 3, 5$ — простые (листья рекурсии)

   (b) Сертификат простоты 7. Первообразный корень $g = 3$. Делители $p-1 = 2 \cdot 3$ — $\underline{2, 3}$

Поэтому сертификат: $(3361, 22, ((2,3),(3,1),(5,1),(7,1))), (7, 3, ((2,1),(3,1)))$. Каждая скобка содержит проверяемое число, первообразный корень и разложение $p-1$ на множители. В сертификат входит сертификат для 7, так как $7|p-1$.

(каноническое) Задача 19

Пусть $L \in \mathsf{NP}$. $w_i \in L$, $L^* \ni w = w_1 ... w_n$. Проверочный предикат для L — $R_0(x, y)$. Определим сертификат $y(w)$. Добавим список позиций l_i начал слов w_i в слове w. Количество слов $n \leqslant |w|$, длина записи числа $O(\log |w|)$ (номер позиции не больше, чем длина слова). Тогда суммарная длина $O(n \log |w|) = O(|w| \log |w|)$. Добавим в сертификат $y(w)$ сертификаты y_i для w_i. Определим $R(w, y) = R_0(w[1, l_1 - 1], y_1) \wedge R_0(w[l_1, l_2 - 1], y_2) \wedge ... \wedge R_0(w[l_n, |w| - 1], y_n) \wedge (l_i < l_{i+1})$ — имеется в виду такой предикат, который дает то же значение, но в его записи явным образом не фигурирует n (например, это предикат, построенный по МТ, проверяющей эти условия). Тогда для слов $w \in L^*$ $R(w, y(w)) = 1$ (по построению $w[l_i, l_{i+1} - 1] \in L$, и y_i — их сертификаты). Пусть $R(w, y) = 1$. Тогда задано разбиение слова $1 < l_2 < l_3 < ... < l_n < n$ на подслова, причем каждое подслово из L, значит, $w \in L^*$.

(каноническое) Задача 20

Докажем, что $\overline{L} \in \mathsf{NP}$. Сертификат — список подразбиений ребер графа $K_{3,3}$ или K_5 (и указание, какого именно графа) + описание соответствий вершин полученного графа и входного. Тогда для не планарных графов (слов-описаний графов из $\overline{L}$) такой сертификат существует (теорема Куратовского). Количество подразбиений не больше, чем количество вершин в исходном графе, каждое кодируется константой символов. Соответствие кодируется $O(|V|)$ символами (V — вершины входного графа). Длина входного слова не меньше, чем $|V|$ и $|E|$. Поэтому длина сертификата $|y(w)| = O(|V| + |V|) = O(|V|) = O(|w|)$. Проверочный предикат: выполняем подразбиения ребер (каждое за константу времени), проверяем соответствие ребер двух графов с заданным соответствием вершин за $O(|E|)$. Время $O(|V| + |E|) = O(|w|^2)$.

Вспомогательные утверждения

1. $\leqslant_m^p$ — транзитивно. Действительно, пусть $\Sigma_1^* \supseteq A \leqslant_m^p B \subseteq \Sigma_2^*$, $B \leqslant_m^p C \subseteq \Sigma_3^*$. Тогда существуют полиномиально-вычислимые функции $f_1 \colon \Sigma_1^* \to \Sigma_2^*$, $f_2 \colon \Sigma_2^* \to \Sigma_3^*$, причем $\forall x \in \Sigma_1^* \, (x \in A \Leftrightarrow f_1(x) \in B)$, $\forall y \in \Sigma_2^* \, (y \in B \Leftrightarrow f_2(y) \in C)$ Фиксируем $x \in \Sigma_1^*$, определим $y = f_1(x)$. Тогда $x \in A \Leftrightarrow \underbrace{f_1(x)}_{y} \in B \Leftrightarrow f_2(f_1(x)) \in C$

   Функция $g(x) \colon \Sigma_1^* \to \Sigma_3^*$ $g = f_2 \circ f_1$ полиномиально-вычислима (как композиция полиномиально-вычислимых). Получаем, что существует полиномиально-вычислимая $g(x)$, такая что $\forall x \in \Sigma_1^* \, (x \in A \Leftrightarrow g(x) \in C)$, откуда $A \leqslant_m^p C$ ■

2. Пусть $A \in \mathsf{NP}$-c, и $A \leqslant_m^p B \in \mathsf{NP}$. Тогда $B \in \mathsf{NP}$-c. Действительно, $A \in \mathsf{NP}$-c $\Rightarrow \forall C \in \mathsf{NP} \hookrightarrow C \leqslant_m^p A$. Фиксируем $C \in \mathsf{NP}$. $A \leqslant_m^p B$, поэтому из 1 следует, что $C \leqslant_m^p B$. Поэтому $\forall C \in \mathsf{NP} \hookrightarrow C \leqslant_m^p B$. Значит, $B \in \mathsf{NP}$-c ■