Лекция 8

Алгоритмически неразрешимые проблемы

1) Проблема самоприменимости

Множество алгоритмов счётно. Зафиксируем их нумерацию $A_0, A_1, \dots$ с возможными повторениями.

Число входов тоже счётно, поэтому без ограничения общности, считаем, что на вход подаётся натуральное число.

Определим функцию $F: \mathbb{N} \to \mathbb{N}$ таким образом:

$$\begin{cases} F(i) = 1, & \text{если } A_i \text{ останавливается на входе } i. \\ F(i) = 0, & \text{иначе} \end{cases}$$

Проблема самоприменимости состоит в вычислении функции F.

<u>Утв</u> F алгоритмически неразрешима при $\forall$ нумерации алгоритмов.

$\square$ Предположим противное: F - вычислима. Тогда $\exists$ МТ M_F, которая вычисляет F.

Пусть $M_F(n)$ - её результат на входе n.

Построим МТ M_F' так:

$$M_F' = \begin{cases} \text{пишет 1 и останавливается, если } M_F(n) = 0 \\ \text{зацикливается, если } M_F(n) = 1. \end{cases}$$

Т.к. M_F' - МТ, то у её алгоритма есть какой-то номер m.

Запустим $M_F'(m)$:

1) Пусть $M_F'(m)$ остановилась, тогда $F(m) = 0$, то есть МТ M_F' не остановилась на входе m.

Противоречие.

2) Пусть $M'_F(m)$ зациклилась. Тогда $f(m)=1$, и M'_F должна остановиться.

Противоречие

$\Downarrow$

F - невычислима $\blacksquare$

2) Проблема остановки

Определим функцию $f_S : \mathbb{N} \to \mathbb{N}$ таким образом:

$$f_S(i,j) = \begin{cases} 1; & A_i \text{ останавливается на входе } j \\ 0; & \text{иначе} \end{cases}$$

Проблема остановки заключается в вычислении функции f_S.

<u>Ут.</u> f_S неразрешима при $\forall$ нумерации алгоритмов.

$\square$ Сначала построим МТ M_w, которая делает следующее:

1) С пустого входа печатает w

2) Смещается на первый символ

3) Запускает M на w.

Построение M_w:

$\delta(q_0; \lambda) = (w_1; +1; q_1)$

$\delta(q_1; \lambda) = (w_2; +1; q_2)$

$\vdots$

$\delta(q_{|w|-1}; \lambda) = (w_{|w|}; 0; q_\lambda)$

$\delta(q_\lambda; a) = (a; -1; q_\lambda)$

$\delta(q_\lambda; \lambda) = (\lambda; +1; q_{0M})$

Здесь w_i - i-ый символ слова w

a - $\forall$ непустой символ

q_{0M} - начальное состояние M. Дальше выполняется её код

Построение M_w свело задачу к вопросу: „остановится ли МТ на пустом входе?"

Возьмём произвольную нумерацию алгоритмов $A_0; A_1 ...$

Пронумеруем входы так, что у A_i i-ый вход – пустой.

Тогда задача свелась к тому, что спрашивается, остановится ли A_i на входе i.

Это неразрешимая проблема самоприменимости для $\forall$ нумерации алгоритмов.

Тогда f_S неразрешима, т.к умея её вычислять, мы бы умели решать проблему самоприменимости ■

<u>Опр</u> L_{stop} – язык в алфавите $\{0;1\}$, состоящий из тех описаний $\langle M; w \rangle$ (МТ M и входного слова w), для которых M останавливается на входе w.

<u>Утв</u> Язык L_{stop} неразрешим

3) <u>Достижимость в ассоциативном исчислении</u>

Заданы правила преобразования слов в алфавите A: $L_i \to R_i \in A^*$ (о нет, отсылка на звездец из алгосов :D)

$L_1 \to R_1; L_2 \to R_2 ... L_n \to R_n$, то есть $\forall$ вхождение L_i в слово можно заменить на R_i.

Задача достижимости: по двум заданным словам и набору правил определить, можно ли одно слово получить из другого заданными преобразованиями.

<u>Утв</u> Задача достижимости неразрешима

□ Построим M' так. Пусть M – МТ, решающая задачу достижимости:

1) Работает как M

2) Из любого финального состояния M делает следующее: стирает „всё" с ленты

(+ GarbageCollector; java moment) и переходит в своё единственное финальное состояние и останавливается

Тогда если бы эта задача была разрешима, то про M' можно было бы всегда сказать, останавливается ли она на входе w, то есть была бы разрешима проблема остановки $\Rightarrow$ задача достижимости неразрешима ■

4) Равенство слов в полугруппе

Заданы правила преобразования слов в алфавите A: $L_i, R_i \in A^*$ (опять...):

$L_1 \leftrightarrow R_1;\ L_2 \leftrightarrow R_2 \ldots L_n \leftrightarrow R_n$, то есть $\forall$ вхождение L_i в слово можно заменить на R_i и наоборот.

Задача достижимости: по двум заданным словам и набору правил определить, можно ли одно слово из другого заданными преобразованиями.

Утв Эта задача достижимости неразрешима.

□ Сделаем $2n$ односторонних замен: $L_1 \to R_1;\ L_2 \to R_2 \ldots L_n \to R_n;\ R_1 \to L_1 \ldots R_n \to L_n$

Получили предыдущую задачу ■

5) Проблема соответствий Поста

Системой соответствий Поста называется набор пар слов в конечном алфавите A: $(u_1; v_1); (u_2; v_2) \ldots (u_n; v_n)$

Решением системы соответствий называется такая конечная последовательность $i_1; i_2 \ldots i_k$, что $u_{i_1} u_{i_2} \ldots u_{i_k} = v_{i_1} v_{i_2} \ldots v_{i_k}$

Утв Задача о $\exists$ решения системы соответствий Поста неразрешима

б) Функция трудолюбия Радо

Функция трудолюбия Радо $R(n)$ определяется как максимальное количество единиц, которое может оказаться на ленте, в результате работы МТ с n состояниями на пустом входе и алфавитом $\{1; \Lambda\}$ при условии, что МТ останавливается на пустом входе.

Вычислим $R(1)$ и $R(2)$:

$R(1) = 0$, потому что $q_0 = q_f$, и МТ сразу остановилась.

$R(2) = 1$:

$Q \backslash A$	Λ	1
q_0		
q_f	–	–

1) $(q_0; \Lambda) = (q_f; \forall; \forall)$ – max одна „1" и сразу останавливается

2) $(q_0; \Lambda) = (q_0; \Lambda; \forall)$ – зациклилась, невозможно

$(q_0; 1; \pm 1)$ – зациклилась, невозможно

$(q_0; 1; 0)$ – единственный вариант. Появилась одна „1".

3) $(q_0; 1) = (q_0; \forall; \forall)$ – зациклилась, т.к. никогда не перейдёт в q_f

$(q_f; \Lambda; \forall)$ – стало 0 „1"

$(q_f; 1; \forall)$ – осталась 1 „1" $\Rightarrow$

$\Rightarrow R(2) = 1$.

Есть оценка $R(3k+3) \geq 4k$, т.е. $\exists$ МТ M_{3k+3}, которая с $3k+3$ состояниями выводит $4k$ „1". (её нам не предъявили, просто возьмём готовую, поверив в её существование)

Утв $R(n)$ невычислима

□ Предположим противное. Тогда $\exists$ МТ M_R, которая её вычисляет. Тогда на вход она получает $\underbrace{11\ldots1}_{n}$, а выводит $\underbrace{11\ldots1}_{R(n)}$. Пусть $c \in \mathbb{N}$ – количество состояний M_R.

Возьмём следующую МТ M':

1) Она с пустого входа работает, как M_{3k+3} и печатает на ленте $4k$ „1".
Использовали $3k+3$ состояния.

2) Переходим на первую „1", не меняя ленты (хватит 3 состояния)

3) Запускаем M_R. То есть в ответе ждём $R(4k)$

Всего состояний $3k+6+c$. При этом мы вывели $R(4k) \Rightarrow R(3k+6+c) \geq R(4k)$.

Т.к. R – ⓗ строго монотонно возрастает, то $4k \leq 3k+6+c$, что неверно при больших k. Противоречие. $\Rightarrow R(n)$ – невычислима ■

P.S. ⓗ – квантор очевидности