Решения домашних заданий

ДЗ 2

Задача 1. Пусть A и B — перечислимые множества. Верно ли, что перечислимыми будут множества $A \cup B$, $A \cap B$, $A \setminus B$?

Верно ли то же самое, но если перечислимость заменить на разрешимость?

Решение.

1. Если A и B разрешимые, то и $A \cup B$, $A \cap B$, $A \setminus B$ тоже разрешимы. Например, $\chi_{A \cap B}(x) = \chi_A(x)\chi_B(x)$ — вычислима, т.к. $\chi_A(\cdot)$, $\chi_B(\cdot)$ — вычислимы. А так как дополнение разрешимого тоже разрешимо, то $A \cup B = \overline{\overline{A} \cap \overline{B}}$ и $A \setminus B = A \cap \overline{B}$ разрешимы.

2. Пусть A и B перечислимы. Тогда можно построить алгоритм, который перечисляет их параллельно (поочередно выдает элементы из A, B). Очевидно, он печатает $A \cup B$.

3. Пусть A и B перечислимы. Опять же, можно параллельно перечислять эти множества, сохраняя все элементы, и печатать те из них, которые встретились в обоих множествах — таким образом, алгоритм напечатает $A \cap B$.

4. Пусть $A = \mathbb{N}$ — перечислимое множество, а B — какое-нибудь перечислимое неразрешимое множество. Тогда $A \setminus B$ — неперечислимое в силу теоремы Поста.

$\square$

Задача 4. Пусть дана некоторая всюду определенная функция $f \colon \mathbb{N} \to \mathbb{N}$. Верно ли, что $f(\cdot)$ является вычислимой, если

1. она неубывающая;

2. она невозрастающая?

Решение.

1. Busy beaver функция является всюду определенной и возрастающей, но она невычислима.

2. Если $f(\cdot)$ невозрастающая, то она будет вычислимой. Действительно, в этом случае ее можно представить в виде $f(n) = c_0 + \sum_{i=1}^{k} c_i \operatorname{Ind}\{n \leq a_i\}$ для каких-то чисел $c_i, a_i \in \mathbb{N}$ (т.к. $f(\cdot)$ "ступенчатая" функция). Таким образом, $f(\cdot)$ является конечной суммой вычислимых функций, поэтому она тоже вычислима.

$\square$

Задача 5. Проверьте перечислимость и разрешимость следующих языков.

1. Язык L_1 состоит из описаний тех МТ, которые останавливаются на *каком-то* входе.

2. Язык L_2 состоит из описаний тех МТ, которые не останавливаются *ни на каком* входе.

Решение. 1. Язык L_1 перечислим. Для всех $n \in \mathbb{N}$ можно с помощью универсальной МТ эмулировать работу первых n машин Тьюринга на первых n входах, делая n шагов. Если за n шагов какая-та МТ остановится на каком-то входе, то печатаем ее описание. Очевидно, таким образом будет перечислен весь язык L_1, т.к. если МТ с номером i останавливается на j-м входе за k шагов, то алгоритм напечатает ее описание на итерации $\max\{i, j, k\}$.

   Покажем, что этот язык неразрешим. Для произвольной МТ M построим МТ M', которая, получив на вход слово x, стирает его, и запускает M на пустом входе. Тогда, если бы существовал алгоритм, решающий, принадлежит ли M' языку L_1, то он решал бы и проблему останова на пустом входе для M, что невозможно.

2. Очевидно, $L_2 = \overline{L_1}$, поэтому он неперечислим по теореме Поста.

□

ДЗ 3

Задача 1. Разрешим ли язык, состоящий из описаний тех МТ, которые останавливаются на *каком-то* входе, используя при этом *менее* 2017 ячеек ленты?

Решение. Этот язык разрешим. Во-первых, заметим, что нас интересуют только входные слова длины менее 2017, которых конечное число. Если длина $|x| \geq 2017$, то достаточно проверить, останавливается ли МТ на первых 2016 символах. Т.к. МТ должна использовать меньше 2017 ячеек ленты, то число ее возможных конфигураций тоже конечно. Это значит, что МТ не останавливается, т.е. зацикливается, если какая-то конфигурация повторилась в ходе работы. Можно составить алгоритм, который, получив на вход описание M, запускает ее на всех входах длины меньше 2017, и при работе на каждом входе сохраняет в памяти все конфигурации МТ. Если на каком-то шаге встретилась конфигурация, которая уже была, то M зациклилась на данном входе. Если же никакая конфигурация не повторяется, то МТ рано или поздно остановится, т.к. конфигураций конечное число. Также, легко проверить, чтобы она использовала менее 2017 ячеек в ходе работы. Таким образом, $\langle M \rangle$ принадлежит языку, только если M остановится на каком-то из этих слов, что можно решить с помощью описанного алгоритма. □

Задача 4. Будем говорить, что состояние q машины Тьюринга M — *недостижимое*, если МТ ни разу не попадает в это состояние ни при каком входе x. Разрешим ли язык L_{unatt}, состоящий из описаний тех МТ, у которых имеется недостижимое состояние?

Решение. Язык L_{unatt} неразрешим, т.к. к нему можно m-свести L_{stop}. Рассмотрим произвольную МТ M. Построим по ней следующую МТ M'. Во-первых, объединим все финальные состояния F в одно состояние *stop*. Во-вторых, добавим новое начальное состояние $start$ и дополнительный символ r. Пусть $Q \setminus F = (q_0, \dots, q_m)$, где q_0 — начальное состояние исходной МТ. Правила перехода следующие:

$$\begin{aligned}
&\delta(start, a) = (start, r, +1),\ a \in \Sigma,\\
&\delta(start, r) = (q_m, r, 0),\\
&\delta(q_i, r) = (q_{i-1}, r, 0),\ 1 \leq i \leq m,\\
&\delta(q_0, r) = (q_0, \sqcup, 0).
\end{aligned}$$

Таким образом, M' стирает входное слово, переходит во все состояния, кроме финального, и запускает M на пустом входе. Следовательно, единственным недостижимым состоянием у M' может быть только $stop$, и если это так, то M не останавливается на пустом входе. Получаем, что

$$M \notin L_{stop} \Leftrightarrow M' \in L_{unatt},$$

то есть $\overline{L_{stop}} \leq_m L_{unatt}$. Т.к. L_{stop} неразрешим, то L_{unatt} тоже неразрешим. □

ДЗ 4

Задача 4. Вспомните алгоритм поиска медианы с семинара. Как вы помните, мы получили следующее рекуррентное соотношение:

$$T(n) \leq \begin{cases} C, & n < n_0, \\ T\left(\frac{n}{5}\right) + T\left(\frac{7n}{10}\right) + cn, & n \geq n_0. \end{cases}$$

1. Покажите (например, по индукции), что решение этой рекурренты $T(n) = O(n)$.

2. Если n не делится на 5, то, вообще говоря, наша рекуррента неверна. Пусть мы определяем размер подзадач с точностью до константы:

   $$T(n) \leq T\left(\frac{n}{5} + A\right) + T\left(\frac{7n}{10} + A\right) + cn,\ n \geq n_0.$$

   Покажите, что асимптотическая оценка сложности не изменится.

3. Как изменится рекуррентное уравнение и оценка сложности, если вместо пятерок на первом шаге алгоритма разбить массив на семерки?

Решение. 2. Нужно показать, что для какой-то константы B выполняется $T(n) \leq Bn$ При всех n. Допустим, что это верно для $1 \leq k \leq n-1$. Тогда при $n \geq n_0$ и $\frac{7n}{10} + A \leq n - 1$, т.е. $n \geq \frac{10(A+1)}{3}$, выполняется

$$T(n) \leq T\left(\frac{n}{5} + A\right) + T\left(\frac{7n}{10} + A\right) + cn \leq B\left(\frac{n}{5} + A + \frac{7n}{10} + A\right) + cn =$$
$$= Bn\left(\frac{9}{10} + \frac{2A}{n} + \frac{c}{B}\right) \leq Bn\left(\frac{9}{10} + \frac{2A}{n_0} + \frac{c}{B}\right).$$

Можно выбрать сколь угодно большое n_0, и рекуррента все равно будет выполняться, но с какой-то другой константой C. Соответственно, при достаточно больших n_0 и B выполняется

$$\frac{9}{10} + \frac{2A}{n_0} + \frac{c}{B} \leq 1,\ C \leq B,$$

поэтому $T(n) \leq Bn$ при всех n, т.е. $T(n) = O(n)$.

3. Если разбивать массив на семерки, то получим следующую рекурренту:

$$T(n) \leq \left(\frac{n}{7}\right) + T\left(\frac{5n}{7}\right) + cn.$$

$\frac{1}{7} + \frac{5}{7} = \frac{6}{7} < 1$, поэтому $T(n) = O(n)$ и в этом случае.

□