Домашнее задание 20

Ткачев Андрей, группа 166

27 февраля 2017 г.

Задача 1. **(a)** Да, верно. Перечислимо $\mathbb{N}$, и вычислимы $\pi(i), \pi(i+1), \ldots, \pi(i+4) \Rightarrow$ перечислитель всех наборов подряд идущих цифр числа π длины 5 может для каждого перечисленного $i \in \mathbb{N}$ вывести $\pi(i), \pi(i+1), \ldots, \pi(i+4)$. Так как каждая последовательность из множества последовательностей входящих в π входит в π с некоторой позиции i, то на i-ой иттерации перечислителя эта последовательность будет выведена. И наоборот, если какая-то последовательность выведена, то она содержится в π и ее длина 5.

(б) Да, верно. Множество всех последовательностей цифр длины 5 конечно (содержит 10^5 элментов), а значит множество последовательностей длины 5, входящих в π является подмножеством конечного множества, а значит конечно $\Rightarrow$ разрешимо.

Задача 2. Заметим, что функция $didgitSum(n) : \mathbb{N} \to \mathbb{N}$ вычислима и всюду определена (для каждого входа, выдает результат за конечное число шагов). Тогда приведем пример перечислителя чисел из X, сумма цифр которых равна 10.

for $x \in X$ **do**
  if $didgitSum(x) = 10$ **then**
    $print(x)$;
  end
end

Algorithm 1: Enumerante Y

Так как каждый элемент X будет перечислен, то для каждого из них будет вычислена сумма цифр, то каждое число из Y будет выведено.

Задача 3. Будем считать, что мы обладаем неограниченным объемом памяти. A и B перечислимы, значит существуют алгоритмы, их перечисляющие. Составим перечислитель $A \times B$. Вывод каждого из перечислителей будем записывать в один из массивов — массив A_{out} и массив для B_{out} (т.е. вместо команды печати будет команда записи в массив). Будем выполнять по-очередно шаги этих перечислителей. После очередного выполненного шага будем выводить все попарные сочетания элементов из массива A_{out} и из массива B_{out} (так, чтобы элементы из A были первыми в паре). Таким образом,для любой пары $(a, b) \in A \times B$ верно, что одно из чисел a, b будет добавленно в соответствующий массив раньше другого, так как будет перечислено соответсвующим перечислителем, и после шага, на котором перечисляется второе число из пары другим перечислителем, в массивах будут оба числа a, b, значит будет выведена пара (a, b), т.е. все пары из $A \times B$ будут перечислены. Более того, очевидно, что перечисляются только парвы вида (a, b), $a \in A,\ b \in B$, значит полученный перечислитель перечисляет ровно $A \times B$.

Задача 4. Пусть $E(f(x)) = \mathbb{N} \backslash M$. Если M — пустое множество, то $f(x) = id_{\mathbb{N}}$, очевидно вычислимая функция. Иначе, пусть $x_0 = sup\ M$ (M конечно, значит точная верхняя грань существует). Поймем, что $\forall x \geq x_0 : f(x) = f(x_0) + x - x_0$ ($f(x)$ — строго возрастает, и начиная с x_0 содержит все натуральные числа, большие $f(x_0)$, т.е. значения функции на соседних аргументах — последовательные натуральные числа). Множество M конечно, значит $\forall x' \in M$ можно запомнить $f(x')$. Таким образом вычисление $f(x)$ сводится к проверке неравенства $x \geq x_0$, и в случае истинны результат — приведенная выше формула, в противном случае ответ — одно из извествных значений $f(x)$ на некотором конечном множестве, т.е. вычислимый ответ. Таким образом, $f(x)$ — вычислима.

Задача 5. Мы знаем, что существует неразрешимое множество $Y \in N$ (т.к. число алгооримов счетно, а подмножеств натуральных чисел — континуум). Мы так же знаем, что $X = \mathbb{N}$ — разрешимое множество (его характеристическая функция — константа, равная 1). Но $X \cup Y = \mathbb{N} \cup Y = \mathbb{N} \Rightarrow X \cup Y$ — разрешимо. Однако, по праву выбора — Y не разрешимо.

Ответ: Да, существуют. Например, $X = \mathbb{N}$, а Y — любое неразрешимое множество.

Задача 6. Мы знаем, что функция $isPrime(x)$, равная 1, если x — простое, и 0 в противном случае, — вычислима и всюду определена(для любого x можно за $x^{\frac{1}{2}}$ операций выяснить простоту — простым перебором чисел от 2 до корня и проверкой x на делимость: x — не просто, если одно из чисел на отрезке $[2, \lfloor\sqrt{x}\rfloor]$ является делителем x). Множество S разрешимо, значит перечислимо. Таким образом, можно предоставить алгоритм, перечисляющий элементы D:

```
for x ∈ S do
    for x_0 ≤ x do
        if isPrime(x_0) then
            print(x_0);
        end
    end
end
```

Algorithm 2: Enumerante D

Данный алгоритм действительно печатает только простые числа, которые являются делителями какого-то из чисел в S, причем кажыдый такой простой делитель будет распечатан, т.к. перечислитель S перечисляет все числа в S, а функция проверки на простоту вычислима.

Задача 7. Приведем алгоритм вычисления f^{-1}:

```
Data: y
Result: f^{-1}(y)
for x ∈ ℕ do
    if y = f(x) then
        print(x); break;
    end
end
```

Algorithm 3: Compute f^{-1}

Так как f — биекция, то $\forall y \in \mathbb{N}\ \exists x \in \mathbb{N} : f(y) = x$, причем $y = f^{-1}(x)$. Таким образом, т.к. $\mathbb{N}$ — перечислимое множество, а $f(x)$ — вычислима, то за конечное число иттераций для конкретного y будет найден $x : f(x) = y$, которой и будет равен $f^{-1}(y)$.