Занятие 8

Три варианта определения термина «перечислимое множество»:
Множество $A \subseteq \Sigma^*$ называется *перечислимым*, если

(Вариант 1) A есть область значений некоторой вычислимой функции.
(Вариант 2) A есть область значений некоторой тотальной вычислимой функции или $A = \emptyset$.
(Вариант 3) A есть область определения некоторой вычислимой функции.

Теорема 1. (Вариант 1) $\Leftrightarrow$ (Вариант 2) $\Leftrightarrow$ (Вариант 3)

Примеры. (Вариант 3) Все разрешимые множества и $\{n \in \mathbf{N} \mid \exists x, y \in \mathbf{N}(n = x^2 - y^2)\}$ перечислимы. Область определения вычислимой функции, которая не имеет тотального вычислимого продолжения, перечислима и неразрешима.

Далее пусть $\Sigma^* = \{0,1\}^* = \{\Lambda, 0, 1, 00, 01, 10, 11, 000, \ldots\}$. Заметим, что

1. Σ^* можно представить в виде пересчета без повторений

   $$\Sigma^* = \{\Lambda, S(\Lambda), S(S(\Lambda)), \ldots\},$$

   где S — тотальная вычислимая функция (действие). Тем самым получим вычислимую в обе стороны биекцию между $\mathbf{N}$ и Σ^*, т.е. вычислимое кодирование всех натуральных чисел всеми двоичными словами.

2. Существует вычислимая в обе стороны биекция между $(\Sigma^*)^2$ и Σ^*, т.е. вычислимое кодирование всех пар двоичных слов всеми двоичными словами. (Такая биекция имеется между $\mathbf{N^2}$ и $\mathbf{N}$, используем ее вместе с S.)

Докажем теорему 1.

3. (Вариант 1) $\Leftrightarrow$ (Вариант 2).

   ($\Leftarrow$) — очевидно, докажем ($\Rightarrow$). Пусть $A = E(f)$, где $f : \Sigma^* \to \Sigma^*$ — вычислимая (частичная) функция. Два случая: $A = \emptyset$ (тривиальный) и $A \neq \emptyset$. Разберем второй. Надо построить тотальную вычислимую функцию $g : \Sigma^* \to \Sigma^*$, для которой $E(g) = E(f) = A$. Фиксируем некоторое слово $a \in A$. Алгоритм вычисления g :

   Вход: слово $x \in \Sigma^*$.

- С помощью вычислимой биекции $\varphi : \Sigma^* \to (\Sigma^*)^2$ вычисляем пару слов $(u, v) := \varphi(x)$ (т.е. декодируем x как пару слов).
- Перебором (в цикле) находим натуральное число n, для которого $S^n(\Lambda) = v$ (т.е. декодируем v как натуральное число).
- Моделируем n шагов вычисления $f(u)$. Если за это время вычисление закончилось, то в качестве результата возвращаем $f(u)$. Если не закончилось, то возвращаем a.

Ясно, что g — тотальная вычислимая функция и $E(g) \subseteq E(f)$. Пусть $b = f(u)$ для некоторого u. Т.к. φ — биекция, то найдется x, для которого $\varphi(x) = (u, S^t(\Lambda))$, где t — точное число шагов вычисления $f(u)$. Для такого x будет $g(x) = f(u) = b$. Тем самым, $E(g) = E(f)$.

4. (Вариант 2) $\Rightarrow$ (Вариант 3).

   Если $A = \emptyset$, то A есть область определения нигде не определенной функции, которая вычисляется алгоритмом `while(1){}` .

   Если $A = E(g)$, где g тотальная вычислимая функция, то зададим вычислимую функция h следующим алгоритмом (неограниченный последовательный перебор):

   Вход: слово $x \in \Sigma^*$.

   - Последовательно вычислять $g(u)$ для $u = \Lambda, S(\Lambda), S(S(\Lambda)), ...$ и сравнивать с x. Если (когда) найдется u, для которого $g(u) = x$, то остановиться и положить $h(x) := 1$.

   Легко видеть, что $D(h) = E(g)$.

5. (Вариант 3) $\Rightarrow$ (Вариант 1).

   Пусть $A = D(h)$ для вычислимой функции h. Переделаем программу h так, чтобы она делала те же вычисления, но вместо вычисленного значения возвращала аргумент: `h(x); return x.` Для модифицированной таким образом функции h' будет $E(h') = D(h') = D(h)$.

Пусть теперь Σ — произвольный алфавит и $A \subseteq \Sigma^*$.
Теорема 2. (A породимо) $\Leftrightarrow$ (A перечислимо).

6. (A породимо) $\Rightarrow$ (A перечислимо).

   Условимся записывать конечные последовательности слов одним словом через разделитель $\# \notin \Sigma$, после чего устроим побуквенное кодирование таких слов словами в алфавите Σ. (Например, каждую букву изобразим своим блоком нулей и единиц; длина блоков фиксирована.) Рассмотрим исчисление общего вида (R, F) в алфавите Σ, которое порождает множество A. Множество D, состоящее из кодов всех его результативных выводов, разрешимо.

   (По слову $x \in \Sigma^*$ пытаемся восстановить последовательность слов $\alpha(x) = v_1, \ldots, v_n$, чьим кодом является x, и с помощью R, F проверяем, является ли $\alpha(x)$ результативным выводом. Возвращаем 1, если является, и 0 в остальных случаях.)

   Зададим вычислимую функцию $f : \Sigma^* \to \Sigma^*$, положив $f(x)$ равным последнему члену последовательности $\alpha(x)$ для $x \in D$ и неопределенным для $x \notin D$. Тогда $E(f) = A$ перечислимо (вариант 1).

7. Пусть перечислимое множество A представлено областью значений функции f, вычислимой с помощью алгоритма Маркова $\mathcal{A}$ без заключительных редукций (без $u \to \cdot v$). Тогда A породимо.

   Пусть функция $f : \Delta^* \to \Delta^*$ вычисляется алгоритмом Маркова $\mathcal{A}$ в алфавите $\Sigma \supseteq \Delta$, причем $\mathcal{A}$ не имеет заключительных редукций. Искомое исчисление в алфавите Σ имеет множество правил

   $$R: \quad \frac{-}{v} \text{ (если } v \in \Delta^*) \qquad \frac{u}{v} \text{ (если } \mathcal{A} \text{ за 1 шаг преобразует } u \text{ в } v)$$

   и условие результативного завершения

   $$F = \{v \in \Delta^* \mid \text{ни одну редукцию } \mathcal{A} \text{ нельзя применить к } v\}.$$

   Множества R и F разрешимы. Завершающиеся вычисления алгоритма $\mathcal{A}$ на входных словах $v \in \Delta^*$ соответствуют минимальным результативным выводам исчисления (тем, из которых нельзя выбросить ни одного промежуточного слова). Каждый результативный вывод можно преобразовать в минимальный с тем же последним словом. Тем самым, исчисление допускает в точности те слова, которые принадлежат области значений функции f.

8. В предыдущей задаче можно избавиться от требования отсутствия заключительных редукций.

   Пусть соответствующая функция $f : \Delta^* \to \Delta^*$ вычисляется алгоритмом Маркова $\mathcal{A}$ в алфавите Σ, ограничений на способы остановки алгоритма $\mathcal{A}$ нет. Рассмотрим некоторый шаг работы алгоритма $\mathcal{A}$ над входным словом. Текущей конфигурацией вычисления (перед шагом) назовем пару, составленную из текущего слова v и редукции (команды) c, выполненной на предыдущем шаге. Текущую конфигурацию условимся записывать в виде одного слова через разделитель: $v\#\ c$. Для начальной конфигурации c пусто. Пусть $v\#\ c \vdash v_1\#\ c_1$ означает, что $v \in \Sigma^*$, c — незаключительная редукция алгоритма $\mathcal{A}$ или пустое слово и $\mathcal{A}$ за один шаг преобразует слово v в v_1, причем использует при этом редукцию c_1. Искомое исчисление задается следующим разрешимым набором правил R:

$$\frac{}{v\#}\ (\text{если } v \in \Delta^*) \qquad \frac{v\#\ c}{v_1\#\ c_1}\ (\text{если } v\#\ c \vdash v_1\#\ c_1)$$

$$\frac{v\#\ c}{v}\ \begin{array}{l}(\text{если } c \text{ — заключительная редукция}\\ \text{или нет редукций, применимых к } v)\end{array}$$

   Условие завершения тривиально: $\Gamma = \Delta^*$.

9. Из Тезиса Чёрча-Маркова следует, что (A перечислимо) $\Rightarrow$ (A породимо).

   Тезис утверждает, что все вычислимые словарные функции могут быть вычислены алгоритмами Маркова. Как следствие, условие вычислимости по Маркову в двух предыдущих задачах можно ослабить до требования вычислимости f. Тем самым в них установлено, что область значений любой вычислимой функции породима.

Под перечислимостью множеств $A \subseteq (\Sigma^*)^2$ можно понимать перечислимость множества $\{\psi(x, y) \mid (x, y) \in A\} \subseteq \Sigma^*$, где $\psi : (\Sigma^*)^2 \to \Sigma^*$ — вычислимая в обе стороны биекция. Графиком функции $f : \Sigma^* \to \Sigma^*$ называется множество $G = \{(x, y) \mid y = f(x)\} \subseteq (\Sigma^*)^2$.

Теорема 3. Функция f вычислима тогда и только тогда, когда ее график G перечислим.

10. (f вычислима) $\Rightarrow$ (G перечислимо).

    Пусть $\varphi = \psi^{-1}$. Определим вычислимую функцию g следующим алгоритмом:

    Вход: слово $u \in \Sigma^*$.

    - Вычислить $(x, y) := \varphi(u)$ и проверить, что $y = f(x)$. Если да, то результат 1, иначе результат не определен.

    Имеем, $D(g) = \{\psi(x, y) \mid (x, y) \in G\}$, т.е. G — перечислимо.

11. (G перечислимо) $\Rightarrow$ (f вычислима).

    Пусть $h : \Sigma^* \to \Sigma^*$ — тотальная вычислимая функция и $E(h) = \{\psi(x, y) \mid (x, y) \in G\}$. Функцию f можно вычислить следующим алгоритмом:

    Вход: слово $x \in \Sigma^*$.

    - Для $z = \Lambda, S(\Lambda), S(S(\Lambda)), \ldots$ вычислять $(u, v) := \varphi(z)$ и сравнивать u с x. Если (когда) обнаружится совпадение $u = x$, то вернуть результат $f(x) := v$ (и остановиться).