

Математическая логика и теория вычислимости

Лекция 11. Вычислимость, разрешимость, перечислимость

Денис Николаевич Москвин

Кафедра математических и информационных технологий
Санкт-Петербургского академического университета

24.04.2017

- 1 Разрешимые и перечислимые множества
- 2 Вычислимые функции
- 3 Неразрешимые и неперечислимые множества

- 1 Разрешимые и перечислимые множества
- 2 Вычислимые функции
- 3 Неразрешимые и неперечислимые множества

- Непустое множество символов называют *алфавитом* Σ .
- *Строка* в некотором алфавите — конечная последовательность символов этого алфавита.
- Множество всех строк обозначают Σ^* .
- Пример. $\Sigma = \{a, b\}$, тогда

$$\Sigma^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

- Непустое множество символов называют *алфавитом* Σ .
- *Строка* в некотором алфавите — конечная последовательность символов этого алфавита.
- Множество всех строк обозначают Σ^* .
- Пример. $\Sigma = \{a, b\}$, тогда

$$\Sigma^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

- Имеется взаимно-однозначное соответствие между строками и натуральными числами:

$$\Sigma^* = \left\{ \underbrace{\Lambda}_0, \underbrace{a}_1, \underbrace{b}_2, \underbrace{aa}_3, \underbrace{ab}_4, \underbrace{ba}_5, \underbrace{bb}_6, \underbrace{aaa}_7, \underbrace{aab}_8, \dots \right\}$$

- Понятие алгоритма формально пока определять не будем.
- Опишем, однако некоторые необходимые нам свойства алгоритмов:
 - 1 Алгоритм получает на вход строку.
 - 2 Алгоритм может либо завершить свое выполнение (terminate), либо не остановиться.
 - 3 Останавливающийся алгоритм выдает строку.
 - 4 Не останавливающийся алгоритм может по ходу работы генерировать вывод (потенциально бесконечный).
 - 5 Алгоритм можно записать как строку в некотором алфавите.
 - 6 Алгоритмы состоят из последовательности шагов, и их можно исполнять пошагово.
- Благодаря соответствию $\Sigma^* \longleftrightarrow \mathbb{N}$ мы можем интерпретировать алгоритмы, как принимающие и возвращающие натуральные числа, а не строки.

- Множество $X \subset \mathbb{N}$ называется *разрешимым*, если существует всегда завершающийся алгоритм A , проверяющий принадлежность произвольного $n \in \mathbb{N}$ этому множеству:
 - 1 $n \in X \leftrightarrow A(n) = 1$;
 - 2 $n \notin X \leftrightarrow A(n) = 0$.

- Множество $X \subset \mathbb{N}$ называется *разрешимым*, если существует всегда завершающийся алгоритм A , проверяющий принадлежность произвольного $n \in \mathbb{N}$ этому множеству:
 - 1 $n \in X \leftrightarrow A(n) = 1$;
 - 2 $n \notin X \leftrightarrow A(n) = 0$.
- Любое конечное множество — разрешимо.
- Множество \mathbb{N} — разрешимо.
- \emptyset — разрешимо.

- Множество $X \subset \mathbb{N}$ называется *разрешимым*, если существует всегда завершающийся алгоритм A , проверяющий принадлежность произвольного $n \in \mathbb{N}$ этому множеству:
 - 1 $n \in X \leftrightarrow A(n) = 1$;
 - 2 $n \notin X \leftrightarrow A(n) = 0$.
- Любое конечное множество — разрешимо.
- Множество \mathbb{N} — разрешимо.
- \emptyset — разрешимо.
- Множество $\{n \mid \text{В десятичной записи } \pi \text{ есть } n \text{ семёрок подряд.}\}$ разрешимо?

- Множество $X \subset \mathbb{N}$ называется *разрешимым*, если существует всегда завершающийся алгоритм A , проверяющий принадлежность произвольного $n \in \mathbb{N}$ этому множеству:
 - ① $n \in X \leftrightarrow A(n) = 1$;
 - ② $n \notin X \leftrightarrow A(n) = 0$.
- Любое конечное множество — разрешимо.
- Множество \mathbb{N} — разрешимо.
- \emptyset — разрешимо.
- Множество $\{n \mid \text{В десятичной записи } \pi \text{ есть } n \text{ семёрок подряд.}\}$ разрешимо?
- Существуют ли неразрешимые множества?

- Множество $X \subset \mathbb{N}$ называется *полуразрешимым*, если существует алгоритм A , который для произвольного $n \in \mathbb{N}$:
 - 1 $n \in X \leftrightarrow A(n) = 1$;
 - 2 $n \notin X \leftrightarrow A(n) = \perp$.
- **Утверждение.** Любое разрешимое множество полуразрешимо.

- Множество $X \subset \mathbb{N}$ называется *перечислимым*¹, если существует алгоритм B , который на нулевом входе $B(0) = \perp$, но возвращает все элементы множества X и только их.
- Подразумевается, что возвращается строка из натуральных чисел, входящих в X , в произвольном порядке и с произвольными задержками.
- Требование незавершимости $B(0) = \perp$ накладывают, чтобы от этого алгоритма не требовалось разрешения вопроса о конечности/бесконечности X .

¹В СССР такое X называлось *рекурсивно перечислимым*. 

- **Утверждение.** Любое перечислимое множество полуразрешимо.
- **Доказательство.** Строим алгоритм A так. Для интересующего нас x запускаем B и ждём x . Если появилось, возвращаем 1. ■
- **Утверждение.** Любое полуразрешимое множество перечислимо.
- **Доказательство.** Строим график зависимости числа шагов алгоритма A от натурального x . Затем обходим координатную сетку по диагоналям. ■
- **Утверждение.** Полуразрешимость и перечислимость это одно и то же.
- **Утверждение.** Любое разрешимое множество перечислимо.

- **Теорема.** Пересечение и объединение двух перечислимых множеств перечислимо.
- **Доказательство.** Запускаем параллельно перичисляющие алгоритмы для одного и другого. После каждого нового результата анализируем имеющиеся конечные множества. ■
- А вот с дополнением это не так!
- **Теорема (Поста).** Если множество X и его дополнение $\mathbb{N} \setminus X$ — перечислимы, то X — разрешимо.
- **Доказательство.** Запускаем параллельно (пошагово) полурешающие алгоритмы для X и $\mathbb{N} \setminus X$. Один из них обязан завершиться. ■

- **Теорема.** Множество $X \subset \mathbb{N}$ перечислимо тогда и только тогда, когда оно является проекцией некоторого разрешимого множества пар $Y \subset \mathbb{N} \times \mathbb{N}$.
(Проекция: $x \in X \leftrightarrow \exists y \langle x, y \rangle \in Y$.)
- **Доказательство.** (\Leftarrow) Тривиально.
(\Rightarrow) Пусть B — алгоритм, перечисляющий X . Запускаем его и формируем пары $\langle x, n \rangle$, где n — номер шага, на котором получен x . **Почему это множество пар разрешимо?**

- **Теорема.** Множество $X \subset \mathbb{N}$ перечислимо тогда и только тогда, когда оно является проекцией некоторого разрешимого множества пар $Y \subset \mathbb{N} \times \mathbb{N}$.
(Проекция: $x \in X \leftrightarrow \exists y \langle x, y \rangle \in Y$.)
- **Доказательство.** (\Leftarrow) Тривиально.
(\Rightarrow) Пусть B — алгоритм, перечисляющий X . Запускаем его и формируем пары $\langle x, n \rangle$, где n — номер шага, на котором получен x . **Почему это множество пар разрешимо?**
Для пары $\langle z, k \rangle$ запускаем B и делаем k шагов. Если на последнем шаге получили z , то 1 иначе 0. ■

- 1 Разрешимые и перечислимые множества
- 2 Вычислимые функции
- 3 Неразрешимые и неперечислимые множества

- Пусть $D \subset \mathbb{N}$.
- Функция $f : D \rightarrow \mathbb{N}$ называется *вычислимой*, если существует алгоритм F , который для произвольного $d \in \mathbb{N}$ ведет себя так:
 - 1 $d \in D \leftrightarrow F(d) = f(d)$;
 - 2 $d \notin D \leftrightarrow F(d) = \perp$.
- **Пример.** Константная функция вычислима.
- **Пример.** Нигде не определенная функция вычислима.
- **Может ли произвольная вычислимая функция быть продолжена до всюду определенной?**

- **Утверждение.** Множество X перечислимо тогда и только тогда, когда оно является областью определения некоторой вычислимой функции.
- **Доказательство.**
 - (\Rightarrow) Полуразрешающий алгоритм для X определяет искомую (характеристическую) функцию.
 - (\Leftarrow) Пусть $f : X \rightarrow \mathbb{N}$ вычислима, тогда имеется алгоритм F , завершающийся в точности на элементах X . Модифицируем его, подменив возвращаемое значение на 1, получим полуразрешающий алгоритм. ■

Вычислимость и перечислимость (2)

- **Утверждение.** Множество X перечислимо тогда и только тогда, когда оно является **множеством значений** некоторой вычислимой функции.
- **Доказательство.**
 - (\Rightarrow) Запускаем полуразрешающий алгоритм для X на некотором x . Если останавливается — возвращаем x .
 - (\Leftarrow) Пусть $f : D \rightarrow X$ вычислима (имеем F), тогда D перечислимо по предыдущей теореме, то есть для него есть перечисляющий алгоритм B . Перечисляющий алгоритм для X строим так: запускаем B , как только он возвращает очередное значение d выполняем над этим значением F и возвращаем результат $F(d) = f(d)$. ■

Вычислимость и перечислимость (3)

- **Утверждение.** Функция $f : D \rightarrow X$ вычислима тогда и только тогда, когда ее график $\Gamma_f = \{(x, f(x)) \mid x \in D\}$ является перечислимым множеством.
- **Доказательство.**
 - (\Rightarrow) Строим полуразрешающий алгоритм для пар (x, y) . Запускаем F на x . Если \perp , то замечательно. Если получили $f(x)$, но $f(x) \neq y$, зацикливаемся. Если $f(x) = y$ возвращаем 1.
 - (\Leftarrow) Запускаем перечисляющий алгоритм для Γ_f . Если хотим вычислить $f(x)$, ждем пары (x, y) и возвращаем y . Если не дождалось, то замечательно. ■

- 1 Разрешимые и перечислимые множества
- 2 Вычислимые функции
- 3 **Неразрешимые и неперечислимые множества**

- Существуют ли неразрешимые и/или неперечислимые множества?

- Существуют ли неразрешимые и/или неперечислимые множества? Да.
- Разрешимых (и перечислимых) множеств — счетное число, поскольку они определены через алгоритмы.
- С другой стороны мощность множества всех подмножеств \mathbb{N} строго больше мощности счетного множества.
- Этот факт доказывается канторовской диагональной конструкцией. (Пишем характеристические функции подмножеств как последовательности нулей и единиц.)
- Существуют ли неразрешимые, но перечислимые множества?

Универсальная функция

- Функция $U : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ называется *универсальной* (для класса **унарных вычислимых функций**), если
 - 1 для каждого n ее сечение

$$U_n : x \mapsto U(n, x)$$

является **унарной вычислимой** функцией;

- 2 все **унарные вычислимые** функции встречаются среди U_n .
- **Утверждение.** Существует бинарная вычислимая функция, являющаяся универсальной для класса унарных вычислимых функций.
 - **Доказательство.** Перенумеруем все алгоритмы (например, по длине). Будем обозначать через $\langle i \rangle$ алгоритм с номером i , а через $\#A$ — номер алгоритма A . Положим

$$U(i, x) = \langle i \rangle(x) \quad \blacksquare$$

- Фактически, универсальная функция — это интерпретатор.

Диагональная функция

- Рассмотрим так называемую *диагональную* функцию $u(n) = U(n, n)$.
- Свойства
 - 1 $u(n)$ является вычислимой функцией;
 - 2 $u(n)$ определена не при всех значениях аргумента (поскольку есть никогда не завершающиеся алгоритмы);
 - 3 $u(n)$ невозможно продолжить до всюду определенной вычислимой функции.

Докажем последнее. Пусть существует всюду определенная $u'(n)$, такая что $u'(n) = u(n)$ всюду, где $u(n)$ определена. Рассмотрим всюду определенную вычислимую функцию

$$d(n) = u'(n) + 1$$

Она вычислима, поэтому есть вычисляющий ее алгоритм D (**всюду останавливающийся**). Пусть $k = \#D$. Рассмотрим

$$u(k) = U(k, k) = d(k) = u'(k) + 1$$

Но $u'(k) = u(k)$. Противоречие. ■

- Рассмотрим область определения диагональной функции $u(n) = U(n, n)$

$$W = \{n \mid \langle n \rangle(n) \neq \perp\}$$

- Множество W перечислимо, как область определения вычислимой функции.

- Рассмотрим область определения диагональной функции $u(n) = U(n, n)$

$$W = \{n \mid \langle n \rangle(n) \neq \perp\}$$

- Множество W перечислимо, как область определения вычислимой функции.
- Множество W **неразрешимо**. Действительно, если бы оно было разрешимым, то мы легко могли бы продолжить u до всюду определенной вычислимой функции.

- Рассмотрим область определения диагональной функции $u(n) = U(n, n)$

$$W = \{n \mid \langle n \rangle(n) \neq \perp\}$$

- Множество W перечислимо, как область определения вычислимой функции.
- Множество W **неразрешимо**. Действительно, если бы оно было разрешимым, то мы легко могли бы продолжить u до всюду определенной вычислимой функции.
- **А можно ли построить пример неперечислимого множества?**

- Рассмотрим область определения диагональной функции $u(n) = U(n, n)$

$$W = \{n \mid \langle n \rangle(n) \neq \perp\}$$

- Множество W перечислимо, как область определения вычислимой функции.
- Множество W **неразрешимо**. Действительно, если бы оно было разрешимым, то мы легко могли бы продолжить u до всюду определенной вычислимой функции.
- **А можно ли построить пример неперечислимого множества?** Да. Теорема Поста.

$$W' = \mathbb{N} \setminus W = \{n \mid \langle n \rangle(n) = \perp\}$$

- Пусть

$$W = \{n \mid \langle n \rangle(n) \neq \perp\}$$

разрешимо, тогда разрешимо (и, следовательно, полурешимо)

$$W' = \mathbb{N} \setminus W = \{n \mid \langle n \rangle(n) = \perp\}$$

- Пусть A — полурешающий алгоритм для W' и $\#A = k$.
- Какому из множеств принадлежит k ?

$$k \in W' \rightarrow \langle k \rangle(k) = \perp \rightarrow A(k) = \perp \rightarrow k \notin W'$$

$$k \in W \rightarrow \langle k \rangle(k) \neq \perp \rightarrow A(k) \neq \perp \rightarrow k \in W' \rightarrow k \notin W$$

Противоречие. ■

- Рассмотрим множество

$$H = \{ \langle n, x \rangle \mid \langle n \rangle(x) \neq \perp \}$$

- Множество H является неразрешимым.

- Рассмотрим множество

$$H = \{(n, x) \mid \langle n \rangle(x) \neq \perp\}$$

- Множество H является неразрешимым.
- Действительно, если бы был разрешающий алгоритм, то запуская его на входах (n, n) сделали бы множество W разрешимым.
- *Проблема остановки* заданного алгоритма на заданном входе является алгоритмически неразрешимой (Алан Тьюринг, 1936).

- Метод доказательства неразрешимости с прошлого слайда легко обобщить.
- Пусть имеются множества $A, B \subset \mathbb{N}$. Множество A *m-сводится* к множеству B , нотация $A \leq_m B$, если существует всюду определенная вычислимая функция f , такая что

$$\forall x (x \in A \leftrightarrow f(x) \in B)$$

- Сведение транзитивно.
- Если $A \leq_m B$ и B — разрешимо (перечислимо), то A — разрешимо (перечислимо).
- Пример. $W \leq_m \mathbb{N}$, поскольку в качестве f мы можем взять $n \mapsto (n, n)$. W неразрешимо, следовательно \mathbb{N} неразрешимо.