ФИВТ МФТИ, весна 2014. Специальность ПМИ.

Краткие заметки по курсу *математическая логика и теория алгоритмов*.
Часть вторая: вычислимость.
(А.Е. Ромащенко).

Заметки написаны для студентов, слушавших лекции курса и посещавших семинары на факультете ИВТ Физтеха. Текст непригоден для использования в качестве самостоятельного учебного пособия, независимого от занятий.

1 Формальное определение понятия *алгоритм*.

Существует несколько эквивалентных определений вычислимости/вычислимых функций, формализующих наше интуитивное представление об алгоритмах. Эквивалентное определение вычислимости можно дать с помощью *машин Тьюринга* (см. [1, 4, 5]), *машин поста* [3], *машин Минского* и *нормальных алгорифмов Маркова* и многих других моделей. В нашем курсе мы пользуемся определением вычислимости по Тьюрингу (см. материалы первого семестра).

Таким образом, для нас *алгоритм* — это машина Тьюринга (программа для машины Тьюринга) с одной лентой. Говорят, что алгоритм (машина Тьюринга) M вычисляет функцию

$$f\colon \{0,1\}^* \to \{0,1\}^*$$

(определённую на некотором подмножестве двоичных слов), если получив на вход слово x из области определения f, машина за конечное число шагов печатает на ленте результат $y = f(x)$ и останавливается, а получив на вход слово x, не принадлежащее области определения f, машина не останавливается. Такая функция f называется *вычислимой.*

Аналогично определяются вычислимые функции $f\colon \mathbb{N} \to \mathbb{N}$ (машина, вычисляющая, такую функцию, получает на вход двоичную запись x и посылает на выход двоичную запись $f(x)$), а также вычислимые функции нескольких аргументов ($f\colon \mathbb{N}^k \to \mathbb{N}$ или $f\colon (\{0,1\}^*)^k \to \{0,1\}^*$).

Отметим, что всякую *программу* можно записать как конечный текст в некотором конечном алфавите. Зафиксируем способ кодировки программ и занумеруем все программы в естественном порядке (более короткие тексты программ будут иметь меньший номер, чем более длинные; тексты одинаковой длины нумеруем в алфавитном порядке). Данную нумерацию программ будем называть *стандартной.* (Продумайте детали такой нумерации программ. Обратите внимание на индексы булевых переменных и номера строк программы — все эти числа нужно записать в конечном алфавите.)

Мы говорим, что функция $E\colon \mathbb{N}^2 \to \mathbb{N}$ осуществляет вычислимое кодирование пар натуральных чисел, если

1. E определена на всех парах натуральных чисел,

2. существуют такие вычислимые функции D_1 и D_2, что для любой пары $(n, m) \in \mathbb{N}^2$
$$D_1(E(n, m)) = n, \text{ и } D_2(E(n, m)) = m.$$

(Вычислимые «декодирующие» функции D_1 и D_2 позволяют извлечь из «кода» пары его первую и вторую компоненты.) Аналогично определяется кодирование $E\colon \mathbb{N}^k \to \mathbb{N}$ наборов из k натуральных чисел для произвольного $k = 3, 4, 5, \ldots$

Функция $f\colon \mathbb{N}^k \to \mathbb{N}$ называется *вычислимой*, если существует алгоритм, который получает на вход k натуральных чисел $x_1, \ldots, x_k$ и возвращает значение $f(x_1, \ldots, x_k)$, если функция определена на данном наборе чисел, и не останавливается, если функция на данном наборе не определена. Это определение требует некоторого уточнения: нужно договориться, как мы будем подавать на вход машине набор из k натуральных чисел. Мы договорились, что формально вход у алгоритма только один; однако этот вход может пониматься как*код* кортежа $(x_1, \ldots, x_k)$ (при некотором заранее фиксированном вычислимом кодировании $E\colon \mathbb{N}^k \to \mathbb{N}$).

2 Вычислимые и невычислимые функции

Утверждение 1 *Существует всюду определенная функция $f\colon \mathbb{N} \to \mathbb{N}$, которая не является вычислимой.*

Первое доказательство: Данное утверждение легко вывести из мощностных соображений. Каждая вычислимая функция вычисляется некоторым алгоритмом. (При этом одна и та же функция может вычисляться многими разными алгоритмами, но пока это для нас не существенно.) В нашем определении алгоритм — это программа для машины Тьюринга. Программа для машины Тьюринга есть конечный текст в алфавите фиксированного размера, так что множество всех программ счетно. Следовательно, и множество вычислимых функций тоже не более чем счетное. С другой стороны, множество всевозможных всюду определенных функций $f\colon \mathbb{N} \to \mathbb{N}$ имеет мощность континуум. Следовательно, не все такие функции вычислимы. Можно даже сказать, что в некотором смысле “почти все” функции $f\colon \mathbb{N} \to \mathbb{N}$ невычислимы.

Второе доказательство: Нам однако будет полезно иметь более конкретное описание некоторых невычислимых функций. Чтобы более явным образом построить невычислимую функцию, нам потребуется вспомнить диагональный метод Кантора — тот самый метод, который позволяет доказать, что счетное множество и множество мощности континуум неравномощны.

Занумеруем некоторым естественным образом все программы для машины Тьюринга. В данном случае не очень важно, как именно мы будем нумеровать программы. Удобно воспользоваться самым естественным способом: упорядочить программы по их длине, а программы равной длины упорядочить по алфавиту. Будем называть такой порядок на программах *естественным*. Программы, выписанные в указанном порядке, мы обозначим $p_0, p_1, p_2, \ldots$.

Теперь рассмотрим следующую бесконечную таблицу. Строки таблицы будут соответствовать программам $p_0, p_1, p_2, \ldots$ Столбцы будут соответствовать натуральным числам $0, 1, 2, \ldots$ В клетке на пересечении n-ой строки и m-го столбца поместим значение $p_n(m)$, т.е., результата, который выдает n-ая программа на входе m. Если n-ая программа на входе m не останавливается, то поместим в данную клетку таблицы специальную пометку **undef**.

Рассмотрим диагональ этой таблицы – последовательность значений

$$p_0(0), p_1(1), \ldots, p_n(n), \ldots$$

Напомним, что в этой последовательности кроме натуральных чисел могут встречаться значения **undef**. Определим следующую "антидиагональную" последовательность a_n:

$$a_n = \begin{cases} p_n(n) + 1, & \text{если программа } p_n \text{ останавливается на входе } n, \\ 0, & \text{иначе.} \end{cases}$$

Таким образом, для любого натурального n значение a_n отличается от $p_n(n)$.

Мы утверждаем, что отображение $f\colon n \mapsto a_n$ невычислимо. В самом деле, пусть существует алгоритм, позволяющий вычислить эту функцию. Тогда его можно записать в виде программы для машины Тьюринга, и эта программа получит какой-то номер в стандартной нумерации. Назовем эту программу p_{n_0}. Но если программа p_{n_0} на любом входе n выдает значение a_n, то

$$p_{n_0}(n_0) = a_{n_0} = p_{n_0}(n_0) + 1 \neq p_{n_0}(n_0).$$

(Отметим, что поскольку программа p_{n_0} останавливается на любом входе, в строке номер n_0 в нашей таблице не может встречаться **undef**). Полученное противоречие показывает, что функция $f\colon n \mapsto a_n$ не может быть вычислимой.

Замечание. Приведенное выше рассуждение доказывает следующий факт: не существует такой всюду определенной вычислимой функции $g\colon \mathbb{N} \to \mathbb{N}$, которая продолжала бы частичную функцию $n \mapsto p_n(n)$. (Проведите это доказательство более подробно!)

3 Разрешимые множества.

Первый вариант определения: Множество $A \subset \mathbb{N}$ называется *разрешимым*, если существует алгоритм, который на каждом входе $n \in \mathbb{N}$ выдает ответ 1 ("*да*"), если $n \in A$, и ответ 0 ("*нет*"), если $n \notin A$.

Второй вариант определения: Множество $A \subset \mathbb{N}$ называется *разрешимым*, если его характеристическая функция

$$\chi_A(a) = \begin{cases} 1, & \text{если } x \in A, \\ 0, & \text{если } x \notin A \end{cases}$$

вычислима.

Отметим, что это не два разных (и даже не два эквивалентных) определения, а две формулировки одного и того же свойства, выраженные разными словами. Определения разрешимости означает, что существовует алгоритм, который по заданному числу проверяет, принадлежит ли оно множеству.

Утверждение 2 *Если множества $A, B \subset \mathbb{N}$ разрешимы, то и множества $A \cup B$, $A \cap B$, $A \setminus B$ также являются разрешимыми. Пустое множество и множество $\mathbb{N}$ разрешимы. Все конечные множества разрешимы.*

Доказательство: Попробуйте доказать это утверждение, не заглядывая в конспект лекций. В случае затруднений см. главу 5.2 в [4].

4 Перечислимые множества.

Основное определение: Множество $A \subset \mathbb{N}$ называется *перечислимым* (*полуразрешимым*), если его полухарактристическая функция

$$\chi'_A(x) = \begin{cases} 1, & \text{если } x \in a \\ \text{неопределено}, & \text{иначе} \end{cases}$$

вычислима.

Известно много эквивалентных определений перечислимых множеств. Вот некоторые из них:

- Множество называется перечислимым, если оно является областью определения некоторой вычислимой функции.
- Множество называется перечислимым, если оно является множеством значений некоторой вычислимой функции.
- Множество называется перечислимым, если оно пусто или является множеством значений некоторой всюду определенной вычислимой функции.

Докажите, что эти три определения эквивалентны основному определению.

Утверждение 3 *Всякое разрешимое множество перечислимо.*

Доказательство: Алгоритм вычисления характеристической функции множества нетрудно переделать в алгоритм вычисления полухарактеристической функции того же множества.

На семинарах также рассматривалось ещё одно (довольно необычное) определение перечислимости.

Утверждение 4 *Множество $A \subset \mathbb{N}$ перечислимо, если и только если существует такое разрешимое множество $B \subset \mathbb{N} \times \mathbb{N}$, что*

$$x \in A \Leftrightarrow \exists y\ (x, y) \in B.$$

(Другими словами, множество A есть проекция разрешимого B на первую координату.)

Вспомните доказательство этого утверждения.

Теорема 1 (Э. Пост) *Множество $A \subset \mathbb{N}$ разрешимо, если и только если оно само и его дополнение $\mathbb{N} \setminus A$ перечислимы.*

Доказательство: (а) Пусть множество A разрешимо. Тогда и дополнение его разрешимо. Мы уже знаем, что из разрешимости следует перечислимость. Таким образом, A и $\mathbb{N} \setminus A$ перечислимы.

(б) Пусть A и $\mathbb{N} \setminus A$ перечислимы. Опишем алгоритм, который по заданному x проверяет, принадлежит ли x множеству A.

Согласно определени. перечислимости, полухарактеристические функции множество A и $\mathbb{N} \setminus A$

$$\chi'_A(x) = \begin{cases} 1, & \text{если } x \in A, \\ \text{не определено}, & \text{если } x \notin A \end{cases}$$

и

$$\chi'_{\bar{A}}(x) = \begin{cases} 1, & \text{если } x \notin A, \\ \text{не определено}, & \text{если } x \in A \end{cases}$$

вычислимы. Получив вход x, мы запустим параллельно процессы вычисления $\chi'_A(x)$ и $\chi'_A(x)$. Мы заранее знаем, что остановится только одно их этих двух вычислений. Дождемся момента остановки. Если остановился первый процесс (и обнаружилось, что $\chi'_A(x) = 1$), то можно сделать вывод, что $x \in A$. Если же остановился второй процесс (и обнаружилось, что $\chi'_{\bar{A}}(x) = 1$), то можно сделать вывод, что $x \notin A$.

Объясните более детально, как можно организовать "параллельную работу" двух вычислительных процессов в машине Тьюринга.

5 Универсальные вычислимые функции

Определение: Функция $U\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ называется *универсальной* для семейства всех вычислимых функций функций из $\mathbb{N}$ в $\mathbb{N}$, если

(а) $U(n, x)$ вычислима, и

(б) для любой функции $f \in \mathcal{C}$ найдется такое $n \in \mathbb{N}$, что $n \in \mathbb{N}$ функция $f(x) = U(n, x)$ для всех x (обе части равенства одновременно не определены или одновременно определены и принимают одно и то же значение).

Теорема 2 (теорема об интерпретаторе) *Существует вычислимая функция двух аргументов* $U\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$*, которая является универсальной для семейства всех вычислимых функций из* $\mathbb{N}$ *в* $\mathbb{N}$*.*

Доказательство этой теоремы состоит в построении *универсальной машины* Тьюринга – такой машины $\mathcal{U}$, которая получает на вход пару натуральных числе (n, x) и возвращает результат работы машины p_n (n-ой машины Тьюринга в стандартной нумерации) на входе x. Если программа p_n не останавливается на входе x, то машина $\mathcal{U}$ на входе (n, x) тоже не останавливается. В наших стандартных обозначениях результат работы $\mathcal{U}$ можно описать как

$$\mathcal{U}(n, x) = p_n(x)$$

(как обычно, обе части равенства одновременно не определены или одновременно определены и принимают одно и то же значение). Обдумайте технические детали данной конструкции — как построить машину Тьюринга, которая умеет моделировать работу любой другой машины Тьюринга.

Вычислимую функцию двух аргументов $U\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$, которая является универсальной для семейства всех вычислимых функций из $\mathbb{N}$ в $\mathbb{N}$, для краткости часто называют просто *универсальной вычислимой функцией*.

На универсальную вычислимую функцию можно смотреть как на универсальный язык программирования: $U(n, x)$ есть результат применения программы номер n к входу x. Универсальность означает, что на данном языке можно запрограммировать алгоритм для описания любой вычислимой функции.

Определение: Универсальная вычислимая функция называется *главной* (или *гёделевой*), если для любой вычислимой функции $V\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ найдётся всюду определённая вычислимая функция $t\colon \mathbb{N}\to\mathbb{N}$, такая что $U(t(n), x) = V(n, x)$ при всех n и x.

Теорема 3 (теорема о компиляторе) *Существует главная (гёделева) универсальная вычислимая функция* $U\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$*.*

Набросок доказательства: Универсальная вычислимая функция $\mathcal{U}(n, x) = p_n(x)$ (где p_n есть n-ая машина Тьюринга из стандартной нумерации) является главной. В самом деле, пусть $V\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$ некоторая вычислимая функция. Тогда её вычисляет некоторый алгоритм (машина Тьюринга) π. Это алгоритм получает на вход пару чисел (n, x), а возвращает значение $V(n, x)$ (не останавливается, если функция V не определена на паре (n, x)).

Для каждого $n\in\mathbb{N}$ рассмотрим следующий алгоритм π_n. Он получает на вход число x, а затем моделирует работу алгоритма π_n на паре (n, x) и возвращает полученный результат. Таким образом, для каждого x

$$\pi_n(x) = \pi(n, x)$$

(обе части равенства одновременно не определены или одновременно определены и принимают одно и то же значение). Заметим, что преобразование из n в π_n есть несложная рутинная процедура — нужно взять текст программы π и заменить чтение первого входа этой программы на использование битов двоичной записи фиксированного x. Эту процедуру "трансляции" можно описать в виде алгоритма. Подумайте над техническими деталями этой конструкции: как именно должен работать "транслятор", преобразующий программу π в программу π_n по заданному n?

Программа π_n есть программа для машины Тьюринга, и у неё есть номер в нашей стандартной нумерации программ. Этот номер мы и объявим значением $t(n)$. Нетрудно проверить, что $t(n)$ вычислима, и $U(t(n), x) = V(n, x)$ при всех n и x.

С программистской точки зрения понятие главной (гёделевой) универсальной вычислимой функции является формализацией идеи "хорошего" языка программирования. Как мы увидим ниже, для главных универсальных вычислимых функций удаётся доказать многие важные свойства, выполненные для "естественных" языков программирования, таких как `Pascal`, `C`, `Python` или, скажем, язык программ для машины Тьюринга.

6 Теорема Райса–Успенского

Теорема 4 (Райс–Успенский) *Пусть $U(n, x)$ некоторая гёделева универсальная вычислимая функция и $\mathcal{F}$ нетривиальное семейство вычислимых функций из $\mathbb{N}$ в $\mathbb{N}$ (*нетривиальное *— значит непустое и не совпадающее со множеством всех вычислимых функций). Обозначим через N множество всех номеров всех функций из $\mathcal{F}$ в нумерации $U(n, x)$, т.е.,*

$$N = \{n \mid \phi(x) = U(n, x) \text{ есть функция из } \mathcal{F}\}.$$

Тогда множество N неразрешимо.

Доказательство: Сначала мы докажем теорему Райса–Успенского не для всех гёделевых универсальных функций, а для нашей стандартной универсальной вычислимой функции $U(n, x) = p_n(x)$ (результат применения n-ой машины Тьюринга к входу x).

Предположим, что множество N разрешимо (существует алгоритм, который по заданному $n \in \mathbb{N}$ определяет, принадлежит ли это число n, т.е., принадлежит ли функция $\phi(x) = U(n, x)$ множеству $\mathcal{F}$). Чтобы привести это предположение к противоречию, напомним, что множество

$$K = \{n \mid U(n, n) \text{ определено}\}$$

неразрешимо. Мы покажем, что из разрешимости множества N следует разрешимость множества K.

Обозначим нигде не определённую функцию ξ_0. Без ограничения общности будем считать, что ξ_0 принадлежит $\mathcal{F}$. Далее, пусть ξ_1 некоторая вычислимая функция, не принадлежащая $\mathcal{F}$ (такая функция найдется, поскольку по условию теоремы множество $\mathcal{F}$ нетривиально). Обозначим $\rho(x)$ алгоритм, вычисляющий функцию ξ_1.

Мы сопоставим каждому $n \in \mathbb{N}$ некоторую программу π_n. Ниже мы приводим псевокод π_n.

`На входе` x `выполняем следующие действия:`

1. `Вычисляем значение` $U(n, n)$`. /* здесь может произойти зацикливание */`
2. `Вычисляем` $y = \rho(x)$`. /* вычисляем` $\xi_1(x)$ `*/`
3. `Возвращаем в качестве ответа полученное` y`.`

Если $n \in K$, то на шаге 1 зацикливания не происходит, и управление успешно передается на шаг 2. Таким образом, программа π_n оказывается эквивалентна программе ρ и вычисляет функцию ξ_1. Если же $n \notin K$, то на любом входе x программа зацикливается на шаге 1, так что программа π_n ни на одном входе не останавливается, т.е., вычисляет нигде не определенную функцию ξ_0.

Найдём номер программы π_n в списке всех программ (для машины Тьюринга) в стандартном порядке. Замечаем, что

$$
\begin{array}{ll}
\text{если } n \in K, & \text{то номер программы } \pi_n \text{ принадлежит } N \\
\text{если } n \notin K, & \text{то номер программы } \pi_n \text{ не принадлежит } N.
\end{array}
$$

Таким образом, если у нас есть алгоритм, разрешающий множество N, то с его помощью можно различить ситуации *программа* π_n *вычисляет функцию* ξ_0 и *программа* π_n *вычисляет функцию* ξ_1. Значит, мы можем понять, принадлежит ли число n множеству K. Мы заключаем, что K разрешимо, что и дает требуемое противоречие.

Мы доказали теорему для одной (самой привычной для нас) главной универсальной вычислимой функции. Теперь мы докажем следующее утверждение: если утверждение теоремы Райса–Успенского выполнено хотя бы для одной главной универсальной вычислимой функции, то оно верно и для всех других главных универсальной вычислимой функции.

Пусть $U(n, x)$ главная универсальная вычислимая функция, для которой теорема уже доказана, а $U'(n, x)$ любая другая главная универсальная вычислимая функция. Обозначим

$$N = \{n \mid \phi(x) = U(n, x) \text{ есть функция из } \mathcal{F}\}$$

и

$$N' = \{n \mid \phi(x) = U'(n, x) \text{ есть функция из } \mathcal{F}\}.$$

Мы уже знаем, что N неразрешимо. Теперь нужно доказать, что множество N' тоже разрешимо.

Поскольку $U'(n, x)$ является универсальной вычислимой функцией, существует такая вычислимая всюду определенная функция $t\colon \mathbb{N} \to \mathbb{N}$, что

$$U(n, x) = U'(t(n), x)$$

для всех n и x. Это значит, что

$$n \in \mathbb{N} \leftrightarrow t(n) \in N'.$$

Если N' разрешимо, то и N тоже должно быть разрешимо: чтобы выяснить, принадлежит ли некоторое число n множеству N, мы сначала вычисляем $t(n)$, а затем проверяем, принадлежит $t(n)$ множеству N'. Так что из неразрешимости N следует неразрешимость N'. Теперь теорема полностью доказана.

7 Теорема Клини о неподвижной точке

Теорема 5 (Клини) *Пусть $U(n, x)$ некоторая гёделева универсальная вычислимая функция и $f\colon \mathbb{N} \to \mathbb{N}$ всюду определенная вычислимая функция. Тогда существует такое число $n_0 \in \mathbb{N}$, что*

$$U(n_0, x) = U(f(n_0), x)$$

(для каждого x обе части равенства одновременно не определены или одновременно определены и принимают одно и то же значение).

Прежде чем доказывать теорему, рассмотрим несколько её следствий.

Следствие 1 *(а) Пусть $U(n, x)$ гёделева универсальная вычислимая функция. Тогда существует такое число n_0, что $U(n_0, x) = n_0$.*

(б) Существует такая машина Тьюринга π, которая на любом входе x печатает двоичное преставление текста своей программы.

Доказательство: (а) Рассмотрим отображение $f\colon \mathbb{N} \to \mathbb{N}$, которое переводит вход n в номер программы π_n, которая на любом входе возвращает число n. По теореме Клини у этого отображения есть неподвижная точка n_0. Это значит, что для любого x

$$U(n_0, x) = U(f(n_0), x) = n_0$$

(первое равенство следует из свойства неподвижной точки, а второе из определения функции f). Пункт (б) доказывается аналогично.

Следствие 2 *С помощью теоремы Клини о неподвижной точке можно получить новое доказательство теоремы Райса–Успенского.*

Доказательство: Пусть $U(n, x)$ некоторая гёделева универсальная вычислимая функция и $\mathcal{F}$ нетривиальное семейство вычислимых функций из $\mathbb{N}$ в $\mathbb{N}$ (*нетривиальное* — значит непустое и не совпадающее со множеством всех вычислимых функций). Обозначим через N множество всех номеров всех функций из $\mathcal{F}$ в нумерации $U(n, x)$, т.е.,

$$N = \{n \mid \varphi(x) = U(n, x) \text{ есть функция из } \mathcal{F}\}.$$

Поскольку $\mathcal{F}$ нетривиально, множество N не может быть пустым и не может совпадать со всем $\mathbb{N}$. Значит можно выбрать некоторое число $n_1 \in N$ и некоторое $n_0 \in N$.

Предположим, что множество N разрешимо. Определим отображение $f\colon \mathbb{N} \to \mathbb{N}$ следующим образом:

$$f(n) = \begin{cases} n_0, & \text{если } n \in N, \\ n_1, & \text{если } n \notin N. \end{cases}$$

Данная функция принимает лишь два разных значения. Если n есть номер вычислимой функции из $\mathcal{F}$, то $f(n) = n_0$ есть номер функции *не* принадлежащей $\mathcal{F}$. И наоборот, если n есть номер вычислимой функции *не* из $\mathcal{F}$, то $f(n) = n_0$ есть номер функции из $\mathcal{F}$. Таким образом, для любого $n \in \mathbb{N}$ функции $\phi(x) = U(n, x)$ и $\psi(x) = U(f(n), x)$ не совпадают друг с другом. Это противоречит теореме Клини.

Первое доказательство теоремы Клини (в программистском стиле): Сначала докажем утверждение теоремы Клини не для любой главной универсальной вычислимой функции U, а для главной вычислимой функции, соответствующей нумерации программ на некотором "естественном" языке программирования.

Итак, зафиксируем некоторый "естественный" язык программирования (это может быть язык программирования для машин Тьюринга или, скажем, язык программирования C). Пусть есть некоторая всюду определённая функция f, преобразущая тексты программ в тексты программ (или, что эквивалентно, преобразущая номера программ в номера программ в естественной нумерации). Теорема Клини утверждает, что найдётся такая программ π, что программы π и $\pi' = f(\pi)$ эквивалентны, т.е., для каждого входа x значения $\pi(x)$ и $\pi'(x)$ одновременно не определены или одновременно определены и равны друг другу. Мы покажем, что такая программа π действительно существует, и даже опишем её текст. Ниже мы выписываем псевдокод, соответствующий нужной нам программе.

0. `На входе` x `выполняем следующие действия:`

1. `subroutine ComputeF(string: y) {...};` /* вместо многоточия нужно подставить описание процедуры, вычисляющей функцию f */

2. `subroutine Interpreter(string: p, z) {...};` /* вместо многоточия нужно подставить описание интрепретатора: данная процедура эмулирует работу программы p (написанной на том же самом языке программирования) на входе z */

3. `string s1,s2,s3,s4;` /* объявление строковых переменных переменных */

4. `s1 := "0. На входе` x `выполняем следующие действия:`
   `1. subroutine ComputeF(string: y) {...};`

```
2. subroutine Interpreter(string: p, z) {...};
3. string s1,s2,s3,s4;
4. s1 := ‘‘#’’ ;
5. s2 := substitute(s2, DirectQuotations, UndirectQuotations);
6. s3 := substitute(s2, ‘‘#’’ , s1);
7. s4 := Interpreter(ComputeF(s3),x);
8. print s4;
9. stop." ;
```
/* записываем в переменную s1 текст всей программы за исключение строки номер 5 (чтобы избежать двусмысленности, вместо кавычек "..." , встречающихся в тексте программы, мы используем “...”); вместо 5-ой строки мы помещаем текст s1:=“#” ; при этом важно, что символ # ранее в тексте программы не использовался */

5. `s2 := substitute(s1, DirectQuotations, UndirectQuotations);` /* заменить в s1 все вхождения “...” на "..." и поместить результат в s2 */

6. `s3 := substitute(s2, "#" , s1);` /* берём содержимое строковой переменной s2, находим в нём первое вхождение символа # и заменяем его на значение переменной s1; теперь в s3 содержится текст данной программы */

7. `s4 := Interpreter(ComputeF(s3),x);` /* преобразуем текст нашей программы с помощью функции f, а затем применяем полученную в результате этого преобразования программу к входу x */

8. `print s4;` /* выводим значение s4 на стандартный выход */

9. `stop.`

Нетрудно проверить (проверьте!), что на любом входе x программы π и $f(\pi)$ выдают одинаковые ответы (или обе не останавливаются).

Итак, мы доказали теорему Клини для некоторой одной главной вычислимой функции U. Теперь докажем, что утверждение теоремы Клини выполняется и для любой другой главной нумерации U'.

Пусть f всюду определённая вычислимая функция, и мы хотим найти такое n_0, что

$$U'(n_0, x) = U'(f(n_0), x)$$

для всех x. Поскольку U является главной универсальной вычислимой функцией, существует всюду определенная вычислмая функция I_1 такая, что

$$U'(n, x) = U(I_1(n), x)$$

(I_1 является “транслятором” с языка U' на язык U). С другой стороны, поскольку U' тоже является главной универсальной вычислимой функцией, существует всюду определенная вычислмая функция I_2 такая, что

$$U(n, x) = U'(I_2(n), x)$$

(I_2 является “транслятором” с языка U на язык U').

Теперь рассмотрим функцию $g(n) = I_1(f(I_2(n)))$ (мы транслируем программу номер n с языка U на язык U', применяем преобразование f, а затем результат транслируем обратно на язык U). Полученная композиция трёх всюду определённых вычислимых функций и сама является всюду определённой и вычислимой.

Воспользуемся утверждением теоремы Клини, уже доказанным для U. Мы заключаем, что существует такое m_0, что

$$U(m_0, x) = U(g(m_0), x)$$

для всех x. Теперь нетрудно проверить (проверьте!), что $n_0 := I_2(m_0)$ является неподвижной точкой для U'. Теорема доказана.

См. также аналогичное рассуждение в главе 5.2 в [1].

Второе доказательство Клини (в математическом стиле): Обозначим $d(n) = U(n, n)$. Функция $d(n)$ вычислима, но не всюду определена. Более того, мы знаем, что не существует всюду определенного вычислимого продолжения функции $d(n)$ (см. замечание в конце раздела 2).

Будем использовать обозначение $n \sim m$ для пар натуральных чисел, которые являются номерами одной и той же вычислимой функции в универсальной функции $U(n, x)$. Более точно, мы пишем $n \sim m$, если

$$U(n, x) = U(m, x)$$

для всех x (как обычно, мы требуем, чтобы обе части равенства были одновременно не определены или одновременно определены и принимают одно и то же значение).

Лемма 1 *Существует такая вычислимая всюду определенная функция $D\colon \mathbb{N} \to \mathbb{N}$, что для любого n, для которого значение $d(n)$ определено, выполнено условие $D(n) \sim d(n)$.*

Формальное доказательство леммы: Рассмотрим функцию

$$V(n, x) := U(U(n, n), x).$$

Функция $V(n, x)$ является вычислимой. Следовательно, согласно определению главной универсальной вычислимой функции существует вычислимая всюду определенная функция $t\colon \mathbb{N} \to \mathbb{N}$ такая, что

$$V(n, x) = U(t(n), x).$$

Эту функцию $t(n)$ и можно взять в качестве $D(n)$. В самом деле, она (а) вычислимая, (б) всюду определенная, и (в) для таких n, для которых $U(n, n)$ определено, $D(n) \sim m = U(n, n)$. Лемма доказана.

Неформальное объяснение доказательства леммы: Покажем, как построить нужную нам функцию $D(n)$ для стандартной главной универсальной функции $U(n, x) = p_n(x)$ (результат применения n-ой машины Тьюринга к входу x).

Сопоставим каждому числу n программу π_n, соответствующую следующему псевдокоду:

```
На входе x выполняем следующие действия:
1. Вычисляем значение m = p_n(n). /* здесь может произойти зацикливание */
2. Вычисляем y = p_m(x).
3. Возвращаем в качестве ответа полученное y.
```

Заметим, что если $U(n, n)$ определено, то программа π_n эквивалентна программе p_m для $m = U(n, x)$ (на всех входах x программы π_n и p_m работают одинаково).

В качестве $D(n)$ возьмем номер программы π_n в стандартной нумерации. Понятно, что отображение $n \mapsto D(n)$ вычислимо и всюду определено. Внимание: некоторые программы π_n не определены на некоторых или даже на всех входах x; однако каждому числу n можно сопоставить текст программы π_n, а значит и номер $D(n)$. Отображение отображение $n \mapsto D(n)$ определено для всех n. По построению программы π_n имеем $D(n) \sim p_n(n)$ для всех n, для которых $p_n(n)$ определено.

Данное рассуждение по существу повторяет формальное доказательство леммы, приведенное выше, для универсальной функции $U(n, x) = p_n(x)$.

Воспользуемся доказанной леммой и рассмотрим функцию $f(D(x))$. Это всюду определенная вычислимая функция. Обозначим её

$$g(x) = f(D(x)).$$

Поскольку функция $g(n)$ вычислима, у неё есть свой номер в универсальной вычислимой функции $U(n, x)$: существует такое число n_0, что

$$U(n_0, x) = g(x)$$

для всех x. Наконец, рассмотрим число $m_0 = D(n_0)$.

Мы утверждаем, что число m_0 и является неподвижной точкой f, т.е., $m_0 \sim f(m_0)$. Чтобы доказать это, достаточно проследить цепочку равенств:

$$m_0 = D(n_0) \sim d(n_0) = U(n_0, n_0) = g(n_0) = f(D(n_0)) = f(m_0)$$

- первое равенство в цепочке: из определения m_0;
- эквивалентность "$\sim$": по Лемме 1;
- второе неравенство: по определение функции d;
- третье неравенство: из определения n_0;
- четвертое неравенство: из определения функции g;
- пятое неравенство: из определения m_0.

Замечание: Мы имеем право воспользоваться Леммой 1, поскольку $U(n_0, x)$ определено для всех x (в том числе и для $x = n_0$); это следует из определения функции g — данная функция всюду определена. Теорема доказана.

8 m-сводимость

Определение. Множество $A \subset \mathbb{N}$ m-сводится к множеству $B \subset \mathbb{N}$, если существует такая вычислимая всюду определенная функция $f\colon \mathbb{N} \to \mathbb{N}$, что

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

Обозначение: $A \leq_m B$. (Говорят также, что функция f сводит множество A множеству B.)

Неформальная интерпретация: $A \leq_m B$ означает, что множество A в некотором смысле устроено "не сложнее" множества A; решение задачи о принадлежности к множеству A можно "свести" к задаче о принадлежности к множеству B.

Основные свойства m-сводимости:

- $A \leq_m A$ для любого $A \subset \mathbb{N}$;
- если $A \leq_m B$ и $B \leq C$, то $A \leq C$;
- если $A \leq_m B$, то $(\mathbb{N} \setminus A) \leq_m (\mathbb{N} \setminus B)$;
- если A разрешимо, $B \neq \emptyset$, $B \neq \mathbb{N}$, то $A \leq_m B$;
- если $A \leq_m B$ и B разрешимо, то и A разрешимо;
- если $A \leq_m B$ и B перечислимо, то и B разрешимо.

Попробуйте доказать эти свойства, не заглядывая в конспект лекций. В случае затруднения см. главы 6.1 в [1].

Определение. Пусть $\mathcal{C}$ некоторое семейство подмножеств $\mathbb{N}$. Множество A называется *m-полным* в $\mathcal{C}$, если $A \in \mathcal{C}$ и для любого $b \in \mathcal{C}$ выполнено $B \leq_m A$.

Интуитивный смысл определения: A является алгоритмически самым сложным множеством в $\mathcal{C}$.

Замечание: В некоторых семействах множеств $\mathcal{C}$ нет ни одного m-полного множества; в некоторых семействах имеется несколько m-полных множеств.

Теорема 6 *Множество*

$$K = \{n \mid \text{программа } p_n \text{ останавливается на входе } n\}$$

является m-полным в классе всех перечислимых подмножеств $\mathbb{N}$.

Доказательство: Пусть множество A перечислимо. Это значит, что его полухарактеристическая функция

$$\chi'_A(x) = \begin{cases} 1, & \text{если } x \in A, \\ \text{не определено}, & \text{если } x \notin A \end{cases}$$

вычислима. Пусть данная полухарактеристическая функция вычисляется программой p_n из стандартной нумерации машин Тьюринга. Тогда

$$x \in A \leftrightarrow \text{машина } p_n \text{ останавливается на входе } x.$$

Таким образом,

$$x \in A \leftrightarrow \langle n, x\rangle \in S.$$

Таким образом, функция $f\colon x \to \langle n, x\rangle$ сводит множество A к множеству S.

Теперь докажем, что A m-сводится к K. Для этого мы рассмотрим отображение, которое сопоставляет числу x программу π_x, соответствующую следующему псевдокоду:

```
На любом входе:
```

1. `Вычисляем значение` $p_n(x)$`. /* здесь может произойти зацикливание */`
2. `Возвращаем в качестве ответа число 1.`

Если $x \in A$, то программа π_x на каждом входе возвращает ответ 1. Если же $x \notin A$, то π_x на любом входе не останавливается.

Обозначим $m(x)$ номер программы π_x в стандартном перечислении машин Тьюринга. Тогда

$$x \in A \leftrightarrow m(x) \in K.$$

Следовательно, отображение $x \mapsto m(x)$ сводит множество A к множеству K. Теорема доказана.

См. также доказательство этой теоремы в главе 6.2 в [1].

Список литературы

[1] Верещагин Н., Шень А. Вычислимые функции - М.: МЦНМО, 1999.

[2] Лавров И.А., Максимова Л.Л. Задачи по теории множеств, математической логике и теории алгоритмов - М.: Физико-математическая литература, 1995.

[3] Успенский В.А. Машина Поста. - М.: Наука, 1988.

[4] Верещагин Н.К, Плиско, В.Е., Успенский В.А. Вводный курс математической логики. М.: 1997.

[5] Колмогоров А.Н., Драгалин А.Г. Математическая логика. - М.: КомКнига, 2006.