Математическая логика и теория вычислимости
Лекция 13. Модели вычислений

Денис Николаевич Москвин

Совместная магистратура JetBrains и ИТМО
Разработка ПО / Software Engineering

19.05.2021

План лекции

1. Машина Тьюринга
2. Примитивно рекурсивные функции
3. Примитивно рекурсивные функции и машины Тьюринга

План лекции

Машина Тьюринга

- Машина Тьюринга (Alan Turing, 1936) состоит из
  1. бесконечной в обе стороны *ленты*, разделенной на *ячейки*;
  2. *управляющего устройства* с конечным числом *состояний*.
- Ячейки ленты заполнены символами некоторого алфавита Σ. Имеется выделенный *пустой* символ, которыми заполнены все ячейки, кроме конечного числа.
- Управляющее устройство
  1. всегда позиционировано на какой-то ячейке ленты;
  2. может двигаться вдоль ленты в обе стороны;
  3. умеет считывать и записывать символ в текущую ячейку.
- Среди множества состояний $q_i \in Q$ устройства выделяют
  1. начальное состояние q_0;
  2. подмножество *заключительных* состояний $F \subset Q$, перейдя в которые машина останавливается.
- Имеются *правила перехода* (таблица переходов), описывающие поведение машины в зависимости от текущего состояния и символа в ячейке ленты.

Работа машины Тьюринга

- Таблица переходов представляет собой частичную (partial) функцию и задает *программу* машины Тьюринга

$$\delta \ : \ Q \times \Sigma \to Q \times \Sigma \times \{L, R, N\}$$

- Обычно инструкции (элементы таблицы переходов) записывают так:

$$q_i a_j \to q_{i'} a_{j'} d_k$$

- Работа машины Тьюринга:
  1. входное слово записывается на пустой ленте;
  2. управляющее устройство позиционируется в его начале;
  3. выполняются переходы в соответствии с таблицей переходов;
  4. если машина остановилась, результатом считается слово, считываемое направо от текущей позиции управляющего устройства.
- Каждая машина Тьюринга задает частичную функцию на словах из Σ. Такие функции называют *вычислимыми по Тьюрингу*.

Машина Тьюринга: пример

- Реализуйте машину Тьюринга, складывающую два последовательно записанных на ленте числа в $(\mathtt{S}, \mathtt{0})$-кодировке, например

$$\texttt{␣SSSS0SSS0␣}$$

- Реализуйте машину Тьюринга, складывающую два последовательно записанных на ленте числа в $(S, 0)$-кодировке, например

$$\sqcup SSSS0SSS0 \sqcup$$

- Таблица переходов:

$$\begin{array}{lcl} q_0 S & \to & q_0 S R \\ q_0 0 & \to & q_1 S L \\ q_1 S & \to & q_1 S L \\ q_1 \sqcup & \to & q_1 \sqcup R \\ q_2 S & \to & q_f \sqcup R \end{array}$$

Версии машин Тьюринга

- Машина Тьюринга с несколькими лентами — не меняет класса вычислимых функций, но может улучшить асимптотику.
- На самом деле двухленточной достаточно, дальше выигрыш не больше логарифмического.
- Можно использовать полубесконечную ленту. Это тоже не меняет класса вычислимых функций.
- Недетерминированные машины Тьюринга — для некоторых пар (состояние, текущий символ на ленте) может быть более одного правила перехода.

- Универсальная машина Тьюринга — это машина Тьюринга, которая моделирует поведение произвольной машины Тьюринга на произвольном входе.
- Удобно работать с двумя лентами: на одной кодируем таблицу переходов моделируемой машины, на другой — входные данные для закодированного алгоритма.
- Явное построение такой машины — дело не очень сложное, но содержит много технических деталей.
- Первое конструктивное доказательство ее существование дано Тьюрингом (Alan Turing, 1947)
- Фактически универсальная машина Тьюринга моделирует поведение универсальной функции $U(n, x)$.

Тезис Тьюринга

- **Тезис Тьюринга.** Всякая вычислимая функция вычислима на машине Тьюринга.
- Это не строгое математическое утверждение, а эмпирическое наблюдение, поскольку нет точного определения «вычислимости вообще».
- Рассматривалось много способов дать «широкое» определение понятия вычислимости:
  - машины Тьюринга и Поста;
  - нормальные алгорифмы Маркова;
  - лямбда-исчисление Чёрча;
  - частично рекурсивные функции Клини.
- Тем не менее все эти попытки приводят к одному и тому же классу вычислимых функций.

План лекции

1. Машина Тьюринга
2. Примитивно рекурсивные функции
3. Примитивно рекурсивные функции и машины Тьюринга

Примитивно рекурсивные функции

- Примитивно рекурсивные функции строят индуктивно из 3 базисных функций с помощью 2 операций.
- Базисные функции:
  - константа 0;
  - функция следования («саксессор») $\mathbf{s}(x) = x + 1$;
  - семейство функций проекции: для каждого k их k штук: $\pi_k^i(x_1, \ldots, x_k) = x_i$.
- Операции:
  - если имеется функция f арности k и семейство функций $g_1, \ldots, g_k$, каждая арности n, то операция *подстановки* задает функцию h арности n

$$h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n))$$

  - операция *примитивной рекурсии* (определим чуть позже).
- Используя саксессор можно определить константы $1 = \mathbf{s}(0)$, $2 = \mathbf{s}(1) = \mathbf{s}(\mathbf{s}(0))$ и т.д.

Проекции и подстановка

- Проекции позволяют «управлять» арностью:

$$\begin{aligned} x &= \pi_1^1(x) \\ x &= \pi_2^1(x, y) \\ x &= \pi_3^2(z, x, y) \\ g(x) &= g(\pi_2^1(x, y)) \end{aligned}$$

- **Пример.** Покажите, что функция примитивно рекурсивна:

$$t(x, y) = f\big(g(x), x, h(y, x, x)\big)$$

Проекции и подстановка

- Проекции позволяют «управлять» арностью:

$$\begin{aligned} x &= \pi_1^1(x) \\ x &= \pi_2^1(x,y) \\ x &= \pi_3^2(z,x,y) \\ g(x) &= g(\pi_2^1(x,y)) \end{aligned}$$

- **Пример.** Покажите, что функция примитивно рекурсивна:

$$t(x,y) = f\big(g(x), x, h\big(y,x,x\big)\big)$$

- Воспользуемся проекциями

$$t(x,y) = f\big(g(\pi_2^1(x,y)), \pi_2^1(x,y), h\big(\pi_2^2(x,y), \pi_2^1(x,y), \pi_2^1(x,y)\big)\big)$$

Операция примитивной рекурсии

- Если имеется функция f арности k и функция g арности $k+2$, то операция *примитивной рекурсии* задает функцию арности $k+1$

$$\begin{array}{rcl} h(x_1, \ldots, x_k, 0) & = & f(x_1, \ldots, x_k) \\ h(x_1, \ldots, x_k, \mathbf{s}(y)) & = & g(x_1, \ldots, x_k, y, h(x_1, \ldots, x_k, y)) \end{array}$$

- **Утверждение.** Сложение примитивно рекурсивно.

$$\begin{array}{rcl} \mathtt{sum}(x, 0) & = & x \\ \mathtt{sum}(x, \mathbf{s}(y)) & = & \mathbf{s}(\mathtt{sum}(x, y)) \end{array}$$

- Если имеется функция f арности k и функция g арности $k+2$, то операция *примитивной рекурсии* задает функцию арности $k+1$

$$\begin{array}{lcl} h(x_1,\ldots,x_k,0) & = & f(x_1,\ldots,x_k) \\ h(x_1,\ldots,x_k,\mathbf{s}(y)) & = & g(x_1,\ldots,x_k,y,h(x_1,\ldots,x_k,y)) \end{array}$$

- **Утверждение.** Сложение примитивно рекурсивно.

$$\begin{array}{lcl} \mathtt{sum}(x,0) & = & x \\ \mathtt{sum}(x,\mathbf{s}(y)) & = & \mathbf{s}(\mathtt{sum}(x,y)) \end{array}$$

- Каковы здесь функции f и g из определения операции примитивной рекурсии?

Операция примитивной рекурсии

- Если имеется функция f арности k и функция g арности $k+2$, то операция *примитивной рекурсии* задает функцию арности $k+1$

$$\begin{array}{lcl} h(x_1,\ldots,x_k,0) & = & f(x_1,\ldots,x_k) \\ h(x_1,\ldots,x_k,\mathbf{s}(y)) & = & g(x_1,\ldots,x_k,y,h(x_1,\ldots,x_k,y)) \end{array}$$

- **Утверждение.** Сложение примитивно рекурсивно.

$$\begin{array}{lcl} \mathtt{sum}(x,0) & = & x \\ \mathtt{sum}(x,\mathbf{s}(y)) & = & \mathbf{s}(\mathtt{sum}(x,y)) \end{array}$$

- Каковы здесь функции f и g из определения операции примитивной рекурсии?

$$f(x) = \pi_1^1(x), \qquad g(x,y,z) = \mathbf{s}(z) = \mathbf{s}(\pi_3^3(x,y,z)).$$

Примеры примитивно рекурсивных функций

- **Утверждение.** Умножение примитивно рекурсивно.

Примеры примитивно рекурсивных функций

- **Утверждение.** Умножение примитивно рекурсивно.

$$\begin{aligned} \mathtt{prod}(x, 0) &= 0 \\ \mathtt{prod}(x, \mathtt{s}(y)) &= \mathtt{sum}(\mathtt{prod}(x, y), x) \end{aligned}$$

Примеры примитивно рекурсивных функций

- **Утверждение.** Умножение примитивно рекурсивно.

$$
\begin{array}{lcl}
\mathtt{prod}(x, 0) & = & 0 \\
\mathtt{prod}(x, \mathbf{s}(y)) & = & \mathtt{sum}(\mathtt{prod}(x, y), x)
\end{array}
$$

- Ниже мы будем использовать умножение и сложение в стандартной инфиксной форме.
- **Утверждение.** Усеченное вычитание примитивно рекурсивно.

Примеры примитивно рекурсивных функций

- **Утверждение.** Умножение примитивно рекурсивно.

$$\begin{array}{lcl} \mathtt{prod}(x, 0) & = & 0 \\ \mathtt{prod}(x, \mathbf{s}(y)) & = & \mathtt{sum}(\mathtt{prod}(x, y), x) \end{array}$$

- Ниже мы будем использовать умножение и сложение в стандартной инфиксной форме.
- **Утверждение.** Усеченное вычитание примитивно рекурсивно.
- Определим сначала «предессор», то есть функцию, которая вычитает 1.

$$\begin{array}{lcl} \mathtt{pred}(0) & = & 0 \\ \mathtt{pred}(\mathbf{s}(y)) & = & y \end{array}$$

Примеры примитивно рекурсивных функций

- **Утверждение.** Умножение примитивно рекурсивно.

$$\begin{array}{lcl} \mathtt{prod}(x,0) & = & 0 \\ \mathtt{prod}(x,\mathbf{s}(y)) & = & \mathtt{sum}(\mathtt{prod}(x,y),x) \end{array}$$

- Ниже мы будем использовать умножение и сложение в стандартной инфиксной форме.
- **Утверждение.** Усеченное вычитание примитивно рекурсивно.
- Определим сначала «предессор», то есть функцию, которая вычитает 1.

$$\begin{array}{lcl} \mathtt{pred}(0) & = & 0 \\ \mathtt{pred}(\mathbf{s}(y)) & = & y \end{array}$$

- Теперь имеем для усеченного вычитания

$$\begin{array}{lcl} x \dot{-} 0 & = & x \\ x \dot{-} \mathbf{s}(y) & = & \mathtt{pred}(x \dot{-} y) \end{array}$$

Примитивно рекурсивные множества

- Множество называют *примитивно рекурсивным*, если его характеристическая функция примитивно рекурсивна.
- Множество называют *примитивно рекурсивным*, если оно является множеством нулей примитивно рекурсивной функции.
- Оба эти определения эквивалентны, если $a(x)$ — характеристическая функция A, то $1 \dot{-} a(x)$ обращается в ноль в точности на элементах множества. Обратно аналогично.
- **Утверждение.** Пересечение, объединение и дополнение примитивно рекурсивных множеств примитивно рекурсивны.

Примитивно рекурсивные множества

- Множество называют *примитивно рекурсивным*, если его характеристическая функция примитивно рекурсивна.
- Множество называют *примитивно рекурсивным*, если оно является множеством нулей примитивно рекурсивной функции.
- Оба эти определения эквивалентны, если $a(x)$ — характеристическая функция A, то $1 \dot{-} a(x)$ обращается в ноль в точности на элементах множества. Обратно аналогично.
- **Утверждение.** Пересечение, объединение и дополнение примитивно рекурсивных множеств примитивно рекурсивны.

$$A \cap B$$

Примитивно рекурсивные множества

- Множество называют *примитивно рекурсивным*, если его характеристическая функция примитивно рекурсивна.
- Множество называют *примитивно рекурсивным*, если оно является множеством нулей примитивно рекурсивной функции.
- Оба эти определения эквивалентны, если $a(x)$ — характеристическая функция A, то $1 \dot{-} a(x)$ обращается в ноль в точности на элементах множества. Обратно аналогично.
- **Утверждение.** Пересечение, объединение и дополнение примитивно рекурсивных множеств примитивно рекурсивны.

$$
\begin{array}{ll}
A \cap B & \qquad a(x)b(x) \\
\mathbb{N} \setminus A &
\end{array}
$$

Примитивно рекурсивные множества

- Множество называют *примитивно рекурсивным*, если его характеристическая функция примитивно рекурсивна.
- Множество называют *примитивно рекурсивным*, если оно является множеством нулей примитивно рекурсивной функции.
- Оба эти определения эквивалентны, если $a(x)$ — характеристическая функция A, то $1 \dot{-} a(x)$ обращается в ноль в точности на элементах множества. Обратно аналогично.
- **Утверждение.** Пересечение, объединение и дополнение примитивно рекурсивных множеств примитивно рекурсивны.

$$
\begin{array}{ll}
A \cap B & a(x)b(x) \\
\mathbb{N} \setminus A & 1 \dot{-} a(x) \\
A \cup B &
\end{array}
$$

Примитивно рекурсивные множества

- Множество называют *примитивно рекурсивным*, если его характеристическая функция примитивно рекурсивна.
- Множество называют *примитивно рекурсивным*, если оно является множеством нулей примитивно рекурсивной функции.
- Оба эти определения эквивалентны, если $a(x)$ — характеристическая функция A, то $1 \dot{-} a(x)$ обращается в ноль в точности на элементах множества. Обратно аналогично.
- **Утверждение.** Пересечение, объединение и дополнение примитивно рекурсивных множеств примитивно рекурсивны.

$$\begin{array}{ll} A \cap B & a(x)b(x) \\ \mathbb{N} \setminus A & 1 \dot{-} a(x) \\ A \cup B & 1 \dot{-} (1 \dot{-} a(x))(1 \dot{-} b(x)) \end{array}$$

Примитивно рекурсивные предикаты

- Принадлежность множеству можно интерпретировать как предикат, поэтому можем говорить о *примитивно рекурсивных предикатах*.
- Например, предикаты равенства $x = y$ и неравенства $x \neq y$ примитивно рекурсивны

$$x = y \quad =$$

Примитивно рекурсивные предикаты

- Принадлежность множеству можно интерпретировать как предикат, поэтому можем говорить о *примитивно рекурсивных предикатах*.
- Например, предикаты равенства $x = y$ и неравенства $x \neq y$ примитивно рекурсивны

$$
\begin{aligned}
x = y \quad &= \quad 1 \dot{-} (x \dot{-} y) \dot{-} (y \dot{-} x) \\
x \neq y \quad &=
\end{aligned}
$$

- Принадлежность множеству можно интерпретировать как предикат, поэтому можем говорить о *примитивно рекурсивных предикатах*.
- Например, предикаты равенства $x = y$ и неравенства $x \neq y$ примитивно рекурсивны

$$\begin{aligned} x = y &= 1 \dot{-} (x \dot{-} y) \dot{-} (y \dot{-} x) \\ x \neq y &= 1 \dot{-} (x = y) \end{aligned}$$

- Условный оператор, то есть функция

$$f(x) = \texttt{if } P(x) \texttt{ then } g(x) \texttt{ else } h(x)$$

примитивно рекурсивен, если примитивно рекурсивны предикат P и функции f и g:

$$f(x) =$$

Примитивно рекурсивные предикаты

- Принадлежность множеству можно интерпретировать как предикат, поэтому можем говорить о *примитивно рекурсивных предикатах*.
- Например, предикаты равенства $x = y$ и неравенства $x \neq y$ примитивно рекурсивны

$$
\begin{aligned}
x = y &= 1 \dot{-} (x \dot{-} y) \dot{-} (y \dot{-} x) \\
x \neq y &= 1 \dot{-} (x = y)
\end{aligned}
$$

- Условный оператор, то есть функция

$$f(x) = \texttt{if } P(x) \texttt{ then } g(x) \texttt{ else } h(x)$$

примитивно рекурсивен, если примитивно рекурсивны предикат P и функции f и g:

$$f(x) = P(x)g(x) + (1 \dot{-} P(x))h(x)$$

- Также примитивно рекурсивны факториал, функции `mod` и `div`, предикат «число n простое» (самостоятельно).

- Ограниченные кванторы, примененные к примитивно рекурсивным предикатам дают примитивно рекурсивные предикаты:

$$\begin{aligned} S(x,z) &= \forall y \leqslant z.\ R(x,y) \\ T(x,z) &= \exists y \leqslant z.\ R(x,y) \end{aligned}$$

- Действительно,

$$S(x,z) = \prod_{y=0}^{z} R(x,y),$$

а произведение можно задать рекурсивно

$$\begin{aligned} \prod_{y=0}^{0} f(x,y) &= f(x,0), \\ \prod_{y=0}^{s(t)} f(x,y) &= f(x,s(t)) \cdot \prod_{y=0}^{t} f(x,y). \end{aligned}$$

Примитивно рекурсивный ограниченный график

- Пусть график функции $f(x)$ примитивно рекурсивен, а ее значения ограничены сверху примитивно рекурсивной $g(x)$. Тогда f примитивно рекурсивна.

$$\Gamma_f = \{(x, f(x)) \mid x \in \mathbb{N}\}$$

- Действительно, рассмотрим характеристическую функцию графика $r(x, y)$, тогда

$$f(x) = \sum_{y=0}^{\infty} r(x, y)y = \sum_{y=0}^{g(x)} r(x, y)y$$

А конечная сумма примитивно рекурсивна (аналогично произведению с предыдущего слайда).

Ограниченный оператор минимизации

- Введем для предиката $P(x)$ *ограниченный оператор минимизации* $\mu x < z.\ P(x)$. Он возвращает наименьший x строго меньший z, для котрого этот предикат выполняется, или z, если такого x нет.
- Пусть функция $g(x)$ и предикат $P(x, y)$ примитивно рекурсивны. Тогда примитивно рекурсивна функция

$$f(x) = \mu y < g(x).\ P(x, y)$$

- Действительно, график $f(x)$ легко описать ограниченными кванторами

$$\begin{array}{rcl} P(x, y) = 1 & \wedge & \forall z < y\ P(x, z) = 0 \quad \vee \\ y = g(x) & \wedge & \forall z < y\ P(x, z) = 0 \end{array}$$

Примитивно рекурсивная нумерация пар

- Хотим по паре $\langle x, y \rangle$ примитивно рекурсивно задавать ее номер $[x, y]$ и, наоборот, по номеру $[x, y]$ находить значения элементов пары $x = p_1([x, y])$, $y = p_2([x, y])$.

Примитивно рекурсивная нумерация пар

- Хотим по паре $\langle x, y\rangle$ примитивно рекурсивно задавать ее номер $[x, y]$ и, наоборот, по номеру $[x, y]$ находить значения элементов пары $x = p_1([x, y])$, $y = p_2([x, y])$.
- Есть много подходящих способов нумерации:

$$[x, y] = (2x + 1)2^y - 1$$
$$[x, y] = 2^x 3^y$$
$$\ldots$$

- Почему $p_1(n)$ и $p_2(n)$ будут примитивно рекурсивны?

Примитивно рекурсивная нумерация пар

- Хотим по паре $\langle x, y\rangle$ примитивно рекурсивно задавать ее номер $[x, y]$ и, наоборот, по номеру $[x, y]$ находить значения элементов пары $x = p_1([x, y])$, $y = p_2([x, y])$.
- Есть много подходящих способов нумерации:

$$[x, y] = (2x + 1)2^y - 1$$
$$[x, y] = 2^x 3^y$$
$$\ldots$$

- Почему $p_1(n)$ и $p_2(n)$ будут примитивно рекурсивны?

$$\begin{aligned} p_1(n) &= \mu x \leqslant n.\ \exists y \leqslant n.\ [x, y] = n \\ p_2(n) &= \mu y \leqslant n.\ \exists x \leqslant n.\ [x, y] = n \end{aligned}$$

Совместная рекурсия

- Пусть унарные функции f и g заданы равенствами

$$\begin{aligned} f(0) &= a \\ g(0) &= b \\ f(s(n)) &= F(n, f(n), g(n)) \\ g(s(n)) &= G(n, f(n), g(n)) \end{aligned}$$

где a и b — числа, а F и G — бинарные примитивно рекурсивные функции. Тогда f и g примитивно рекурсивны.

- **Доказательство.**

$$\begin{aligned} h(0) &= [a, b] \\ h(s(n)) &= \Big[F\big(n, p_1(h(x)), p_2(h(x))\big), \\ &\quad\; G\big(n, p_1(h(x)), p_2(h(x))\big)\Big] \end{aligned}$$

$$f(x) = p_1(h(x)), \qquad g(x) = p_2(h(x)). \quad \blacksquare$$

- Пусть $g(x)$ и $h(x, y)$ — унарная и бинарная примитивно рекурсивные функции, причем $g(x) < x$ при положительных x. Тогда унарная функция f, заданная равенствами
$$\begin{aligned} f(0) &= a \\ f(n) &= h(n, f(g(n))) \end{aligned}$$
примитивно рекурсивна.
- **Доказательство.** (см. Верещагина и Шеня)
- Мораль: при определении примитивной рекурсии можно использовать не только предыдущее, но и любое меньшее значение.

План лекции

Теорема

- **Теорема.** Пусть f — вычислимая на машине Тьюринга функция, и время ее работы на входе x не превышает $g(x)$, где $g(x)$ — некоторая примитивно рекурсивная функция. Тогда $f(x)$ — примитивно рекурсивная функция.
- **Доказательство.**

$\blacksquare$