Математические основы Computer Science
Часть 1: Теория алгоритмов. Лекции 2-3.

Дмитрий Ицыксон

ПОМИ РАН

27 сентября 2009

Содержание лекции

1. Теорема Успенского-Райса.
2. Теорема Клини о неподвижной точке.
3. Машины Тьюринга.
4. Предикатные формулы и интерпретации
5. неразрешимость исчисления предикатов.
6. Выразимость в арифметике. Кодирование конечных последовательностей.

В прошлый раз...

- Множество $S \subset \mathbb{N}$ называется разрешимым, если существует такой алгоритм $\mathcal{A}$, что
  - $\forall x \in S, \mathcal{A}(x) = 1$;
  - $\forall x \notin S, \mathcal{A}(x) = 0$.
- Множество $S \subset \mathbb{N}$ называется перечислимым, если существует такой (полуразрешающий) алгоритм $\mathcal{A}$, что
  - $\forall x \in S, \mathcal{A}(x) = 1$;
  - $\forall x \notin S, \mathcal{A}(x)$ не останавливается.
- Множество $S \subset \mathbb{N}$ называется перечислимым, если существует такой (перечисляющий) алгоритм $\mathcal{B}$, который на пустом входе выводит через запятую все элементы множества S.
- Функция $f : M \to \mathbb{N}$, $M \subset \mathbb{N}$ называется вычислимой, если существует такой алгоритм $\mathcal{A}$, что
  - $\forall x \in M, \mathcal{A}(x) = f(x)$;
  - $\forall x \notin M, \mathcal{A}(x)$ не останавливается.

Теорема Успенского-Райса

- $\mathcal{A} \sim \mathcal{B}$, если
  - $\forall x$ $\mathcal{A}(x)$ останавливается $\iff$ $\mathcal{B}(x)$ останавливается;
  - $\forall x$ если $\mathcal{A}(x)$ останавливается, то и $\mathcal{A}(x) = \mathcal{B}(x)$.
- $a \equiv b \iff <a> \sim <b>$.
- Определение. $S \subseteq \mathbb{N}$ называется инвариантным, если $\forall a \in S, b \in \mathbb{N} \setminus S, a \not\equiv b$.
- Теорема. (Успенский, Райс) Если инвариантное S разрешимо, то $S = \emptyset$ или $S = \mathbb{N}$

Доказательство

Теорема. (Успенский, Райс) Если инвариантное S разрешимо, то $S = \emptyset$ или $S = \mathbb{N}$

Доказательство.

- Λ — не останавливающийся алгоритм.
- Пусть $\sharp\Lambda \in S$ и $a \in \mathbb{N} \setminus S$.
- Пусть W — перечислимое неразрешимое множество.
- $V(n,x) = \begin{cases} <a>(x), n \in W \\ \text{не определено, иначе} \end{cases}$
- $V(n,x)$ — вычислимая функция.
- $V_n(x) = V(n,x)$ — вычислимая функция при всех n.
- $V_n = \begin{cases} <a>, n \in W \\ \Lambda, n \notin W \end{cases}$
- $n \in W \iff \sharp V_n \notin S \iff a \notin S$.

О продолжении вычислимых функций

- Мы показали, что не любую вычислимую функцию можно вычислимо доопределить до всюду определенной.
- Лемма. f — вычислимая функция. Тогда $\exists$ всюду определенная вычислимая функция g, которая является $\equiv$-продолжением f ($f(x)$ определено $\Longrightarrow$ $f(x) \equiv g(x)$).
  Доказательство.
  - Алгоритм $\mathcal{A}(n, x)$
    - $k := f(n)$;
    - return $<k>(x)$
  - $\mathcal{A}_n = \mathcal{A}(n, \cdot)$
  - $g(n) = \sharp\mathcal{A}_n$ — всюду определенная вычислимая функция.
  - Если $f(n)$ определена, то $\mathcal{A}_n \sim <f(n)> \Longrightarrow g(n) = \sharp\mathcal{A}_n \equiv f(n)$.

Теорема о неподвижной точке

Теорема. (Клини) h — всюду определенная вычислимая функция. Тогда $\exists m \in \mathbb{N}$, что $m \equiv h(m)$.

Доказательство. От противного.

- $u(n) = <n>(n)$
- $b(n)$ — $\equiv$-продолжение u
- Пусть $t(n) = h(b(n))$ — всюду определенная функция.
- $u(\sharp t) = t(\sharp t) = h(b(\sharp t)) \not\equiv b(\sharp t) \equiv u(\sharp t)$

Программа, печатающая свой текст

- $h : x \longrightarrow \sharp$ “print x”
- m — неподвижная точка.
- $<m>$ печатает m.
- напечатать два раза, второй раз в кавычках, такой текст: “напечатать два раза, второй раз в кавычках такой текст:”
- Бейсик:
  10 LIST

Другое доказательство теоремы о неподвижной точке

- Достаточно доказать теорему для любого языка программирования.
- Пусть в нашем языке есть две встроенные функции:
  - GetProgramText();
  - Execute(s);
- Compute_h(s) — процедура, вычисляющая h.
- Неподвижная точка:
  1. s:=GetProgramText();
  2. s:=Compute_h(s);
  3. Execute(s);

Другое доказательство теоремы Успенского-Райса

Теорема. (Успенский, Райс) Если инвариантное S разрешимо, то $S = \emptyset$ или $S = \mathbb{N}$

Доказательство.

- Пусть S — нетривиальное инвариантное множество.
- $\sharp\Lambda \in S$, $\sharp V \notin S$.
- $h : x \mapsto \begin{cases} \sharp V, x \in S \\ \sharp\Lambda, x \notin S \end{cases}$ — всюду определенная вычислимая функция.
- h не имеет неподвижных точек

Модели вычислений

Зачем они нужны?

- Математическое определение понятия алгоритм;
- Строгое определение сложности алгоритма;
- Возможность доказывать алгоритмическую неразрешимость естественных задач.

Какие Вы знаете модели вычисления?

- λ-исчисление, машина Тьюринга, PAM-машина, машина Поста, нормальные алгоритмы Маркова, программы с конечным числом переменных...
- Почти любой язык программирования может выступать в роли модели вычисления, если есть возможность использовать неограниченное количество памяти.

Машина Тьюринга

- Бесконечная в одну сторону лента, разделенная на ячейки. В самой левой ячейке написан символ $\triangleright$.
- Q — конечное множество состояний. $q_0 \in Q$ — начальное состояние. $q_f \in Q$ — конечное состояние.
- Σ — алфавит символов, которые могут быть записаны на ленте. $\triangleright$, " _ " $\in \Sigma$.
- Головка машины указывает на одну из ячеек ленты
- Правило перехода: $\delta : \Sigma \times Q \to \Sigma \times Q \times \{\rightarrow, \leftarrow, \bullet\}$
- Согласно правилу перехода машина по символу, на который указывает головка, и по текущему состоянию, пишет на это место новый символ, переходит в новое состояние и, возможно, сдвигает головку на 1 символ влево или вправо.
- Начинает работу в состоянии q_0, головка указывает на первый символ. Заканчивает в состоянии q_f.

Пример

Заменить число в двоичной системе счисления на его остаток при делении на 2.

- $(q_0, \frac{0}{1}) \mapsto (q_0, \frac{0}{1}, \rightarrow)$;
- $(q_0, \_) \mapsto (q_1, \_, \leftarrow)$;
- $(q_1, 0) \mapsto (q_2, \_, \leftarrow)$;
- $(q_1, 1) \mapsto (q_3, \_, \leftarrow)$;
- $(\frac{q_2}{q_3}, \frac{0}{1}) \mapsto (\frac{q_2}{q_3}, \_, \leftarrow)$;
- $(\frac{q_2}{q_3}, \triangleright) \mapsto (\frac{q_2}{q_3}, \triangleright, \rightarrow)$;
- $(q_2, \_) \mapsto (q_f, 0, \bullet)$;
- $(q_3, \_) \mapsto (q_f, 1, \bullet)$.

Пример

Делится ли число в 2-ой системе счисления на 3?

- $(q_0, 0) \mapsto (q_0, 0, \rightarrow)$;
- $(q_0, 1) \mapsto (q_1, 1, \rightarrow)$;
- $(q_1, 0) \mapsto (q_2, 0, \rightarrow)$;
- $(q_1, 1) \mapsto (q_0, 1, \rightarrow)$;
- $(q_2, 0) \mapsto (q_1, 0, \rightarrow)$;
- $(q_2, 1) \mapsto (q_2, 1, \rightarrow)$;
- $(q_0, \_) \mapsto (q_{yes}, \_, \bullet)$;
- $(q_1, \_) \mapsto (q_{no}, \_, \bullet)$;
- $(q_2, \_) \mapsto (q_{no}, \_, \bullet)$.

Машина Тьюринга

- Вход машины Тьюринга — то, что записано на ленте. За входом следует бесконечное число пробелов.
- Выход машины Тьюринга — то, что записано на ленте после того, как машина пришла в конечное состояние.
- Если машина Тьюринга проверяет принадлежность языку, то удобно иметь два конечных состояния: q_{yes} и q_{no}.
- Машина Тьюринга может:
  - закончить работу;
  - работать бесконечно.

Сложностные параметры

Время

- Временем работы машины Тьюринга на входе x называем количество шагов, которое машина делает, чтобы прийти в конечное состояние.
- Временной сложностью машины Тьюринга называем максимум по всем входам длины n времени работы машины на этих входах.

Память

- Сложностью по памяти работы машины Тьюринга на входе x называем количество ячеек, в которых побывала головка машины.
- Емкостной сложностью машины Тьюринга называем максимум по всем входам длины n сложности по памяти работы машины на этих входах.

Многоленточная машина Тьюринга

- Есть k лент, головка есть на каждой ленте.
- Правило перехода: $\delta : \Sigma^k \times Q \to \Sigma^k \times Q \times \{\rightarrow, \leftarrow, \bullet\}^k$
- По любой многоленточной машине Тьюринга можно построить одноленточную машину Тьюринга, вычисляющую ту же функцию. Причем сложностные характеристики этой машины будут лишь в полином раз хуже.
- Иногда удобно считать, что машина снабжена специальной входной лентой, доступной только для чтения и выходной лентой, доступной для записи, но без исправлений.

Тезис Черча-Тьюринга

Любой алгоритм можно реализовать в виде машины Тьюринга.

Язык предикатных формул

- Γ — бесконечное множество предметных переменных. $\Gamma = \{x_1, x_2, x_3, \dots\}$.
- $\mathfrak{F} = \{f_1^{(i_1)}, f_2^{(i_2)} \dots\}$ — множество функциональных символов с указанием их арности, $i_k \geq 0$
- Нульместные функциональные символы обычно называют константами.
- Определение. Термы:
  - Предметная переменная $x \in \Gamma$ — терм.
  - Если $f^{(i)} \in \mathfrak{F}$, а $t_1, t_2, \dots, t_i$ — термы, то $f^{(i)}(t_1, t_2, \dots, t_i)$ — терм.
- Пример.
  - $f^{(0)}()$ — терм;
  - $f^{(2)}(x, y)$ — терм;
  - $f^{(2)}(g^{(1)}(x), h^{(3)}(x, y, g^{(1)}(x)))$ — терм.

Язык предикатных формул

- $\mathfrak{P} = \{p_1^{(i_1)}, p_2^{(i_2)} ...\}$ — множество предикатных символов с указанием их арности, $i_k \geq 0$ (в этом множестве есть бесконечное число предикатных символов любой арности).
- Определение. Атомарной формулой называется строчка вида $p^{(i)}(t_1, t_2, \ldots, t_i)$, где $p^{(i)} \in \mathfrak{P}$, а $t_1, t_2, \ldots, t_i$ — термы.
- Определение. Предикатная формула 1-го порядка
  - Если A — атомарная формула, то A — предикатная формула.
  - Если A, B — предикатные формулы, то $(A), \neg A, A \vee B, A \wedge B, A \to B$ — предикатные формулы.
  - Если A — предикатная формула, $x \in \Gamma$, то $\forall x A$ и $\exists x A$ являются предикатными формулами.
- $(\mathfrak{P}, \mathfrak{F})$ — сигнатура формулы.

Примеры предикатных формул

- $p(f(x))$: свободная переменная x;
- $p_1(f_1(x)) \vee p_2()$: свободная переменная x;
- $\forall x \exists y (p_1(z) \vee p_1(x))$: свободная переменная z;
- $\forall x (p_1(f(x))) \to \exists y p_1(y)$: замкнутая формула;
- $\forall y (p_1(x, y) \vee \exists z p_2(f(x, y), g(x)))$: свободная переменная x.

Определение. Переменная называется свободной, если она не входит в область действия квантора по этой переменной. Формула без свободных переменных называется замкнутой.

Интерпретация

- φ — предикатная формула со свободными переменные $x_1, x_2, \ldots, x_k$.
- Интерпретация
  - Носитель интерпретации: множество M
  - $f^{(i)} \in \mathfrak{F}$: отображение $M^i \to M$
  - $p^{(i)} \in \mathfrak{P}$: предикат $M^i \to \{0, 1\}$
  - Переменной x_l сопоставляется элемент M
- В заданной интерпретации можно посчитать значение формулы.
- Моделью называется интерпретация, в которой значение формулы равняется 1.

Примеры

- $\forall x(p(x) \to q(x))$.
  - В интерпретации $M = \mathbb{Z}$, $p(x) = x \vdots 4, q(x) = x \vdots 2$ значение формулы 1.
  - В интерпретации $M = \mathbb{Z}$, $p(x) = x \vdots 3, q(x) = x \vdots 2$ значение формулы 0.
- $\forall x(p(f(x))) \to \forall x p(x)$
  - В интерпретации $M = \mathbb{Z}$, $p(x) = x \vdots 2, f(x) = 2x$ значение формулы 0.

Выполнимость, общезначимость, противоречивость

- Определение. Предикатная формула называется выполнимой, если существует такая интерпретация, при которой значение формулы равняется 1.
- Определение. Предикатная формула называется невыполнимой (или противоречивой), если при всех интерпретациях значение формулы равняется 0.
- Определение. Предикатная формула называется общезначимой (или тавтологией), если при всех интерпретациях значение формулы равняется 1.

Предикат равенства

- Инфиксная запись: пишем $x = y$ вместо $=(x, y)$
- Чтобы во всех интерпретациях он воспринимался одинаково, нужны аксиомы равенства.

  Аксиомы равенства:
- $\forall x \forall y (x = y \to y = x)$ — симметричность
- $\forall x \forall y \forall z ((x = y \wedge y = z) \to x = z)$ — транзитивность
- Для каждого функционального символа $f^{(r)}$:
  $\forall x_1 \ldots \forall x_r \forall y_1 \ldots \forall y_r ((x_1 = y_1 \wedge \cdots \wedge x_r = y_r) \to$
  $f(x_1, \ldots, x_r) = f(y_1, \ldots, y_r))$ — согласованность с функциональными символами
- Для каждого предикатного символа $p^{(r)}$:
  $\forall x_1 \ldots \forall x_r \forall y_1 \ldots \forall y_r ((x_1 = y_1 \wedge \cdots \wedge x_r = y_r) \to$
  $(p(x_1, \ldots, x_r) \to p(y_1, \ldots, y_r)))$ — согласованность с предикатными символами
- Формулу φ с предикатом равенства надо воспринимать как $(A_1 \wedge \cdots \wedge A_n) \to \varphi$

Алгоритмическая неразрешимость

Теорема. Множество тавтологий является неразрешимым.

Доказательство. Сведем задачу об остановке МТ к проверке, является ли формула тавтологией.

- Для каждого ленточного символа $s \in \Sigma$ заводим константу $s()$, для каждого состояния $q \in Q$ заводим константу $q()$.
- Ленту будем кодировать так: $q()|c_1() \circ c_2() \circ \cdots \circ c_{l-1}() \circ g(c_l()) \circ \cdots \circ c_m()$
- Правило $(q_1, c_1) \mapsto (q_2, c_2, \leftarrow)$ записываем формулой: $\forall x \forall y (q_1()|x \circ c_0() \circ g(c_1()) \circ y = q_2()|x \circ g(c_0()) \circ c_2() \circ y)$
- Предикат остановки $stop$: $\forall x (stop(q_f|x))$ для конечного состояния q_f
- МТ остановится на входе $x \iff$ формула $(A_1 \wedge \cdots \wedge A_n) \to stop(q_0()|x_1() \circ \cdots \circ x_m()))$ является тавтологией. A_i — это аксиомы равенства и правила, задающие МТ.

Перечислимость множества тавтологий

- Считаем, что $\mathfrak{F}$ и $\mathfrak{P}$ разрешимы.
- (Теорема Геделя о полноте исчисления предикатов.) Множество тавтологий в сигнатуре $(\mathfrak{F}, \mathfrak{P})$ перечислимо.
- Система доказательств – это алгоритм $\mathcal{A}(\varphi, s)$
  - Если φ — не тавтология, то $\forall s \mathcal{A}(\varphi, s) = 0$;
  - Если φ — тавтология, то $\exists s \mathcal{A}(\varphi, s) = 1$.
- Существуют системы доказательств.
- Алгоритм Британского музея. Перебираем все строчки и проверяем, являются ли они доказательством формулы φ. Если является, то выдать 1.

Выразимые предикаты

- $(\mathfrak{F}, \mathfrak{P})$ — сигнатура.
- Задана интерпретация: M — носитель.
- $P : M^k \to \{0, 1\}$ — некоторый предикат.
- P называется выразимым, если существует формула ϕ с k свободными переменными, что значение ϕ в данной интерпретации при всех оценках свободных переменных совпадает со значением P на этих оценках.

Выразимость в арифметике

- $\mathfrak{F} = \{+, \times\}$, $\mathfrak{P} = \{=\}$
- Носитель интерпретации: $\mathbb{N} = \{0, 1, 2, \dots\}$
- Предикат, выразимый в этой интерпретации называется арифметичным.

  Примеры арифметичных предикатов
- $x \leq y$ $\exists z(x + z = y)$
- $x = 0$ $\forall y(x \leq y)$
- $x = 1$ $\forall y(x \times y = y)$
- $x = k$ $\exists x_1 \dots x_k (x_1 = 1) \wedge \dots \wedge (x_k = 1) \wedge (x = x_1 + \dots + x_k)$
- $x \vdots y$ $\exists z(x = y \times z)$
- x — простое число

  $\neg(x = 1) \wedge ((x \vdots y) \implies ((y = 1) \vee (x = y)))$

Арифметичные предикаты

- $r = a \bmod b$ $\exists q(a = b \times q + r \wedge (r < b))$
- $d = \text{НОД}(a, b)$
- $d = \text{НОК}(a, b)$
- $\text{НОД}(a, b) = 1$
- x — степень двойки
  $\forall y((x \vdots y) \to ((y = 1) \vee \exists z(y = z + z)))$
- x — степень тройки аналогично
- x — степень четверки: x — степень двойки и точный квадрат.
- x — степень 6?

Кодирование конечных последовательностей

Лемма. Для любого целого k найдется сколь угодно большое b, что $b+1, 2b+1, \ldots, kb+1$ — попарно взаимно простые числа.
Доказательство. Общий делитель любых двух чисел — делитель lb, где $1 \leq l \leq (k-1)$. Выберем $b \vdots k!$, тогда любой общий делитель должен быть делителем b, чего не может быть.

Лемма. Для любой последовательности $x_0, x_1, \ldots, x_n$ натуральных чисел можно найти такие числа a и b, что $x_i = a \bmod b(i+1)+1$.
Доказательство. Следует из китайской теоремы об остатках.

- $\exists <x_0, x_2, \ldots, x_n> (\forall i \leq n)[\ldots x_i \ldots]$
- $\exists a, b, n \forall i (i \leq n) \rightarrow [\ldots a \bmod b(i+1)+1 \ldots]$
- $\beta(a, b, i) = a \bmod b(i+1)+1$ — β-функция Геделя.

Арифметичные предикаты

- x — степень 6.
- $\exists a, b, n(\beta(a, b, 0) = 1) \wedge \forall i(i + 1 \leq n) \rightarrow (\beta(a, b, i + 1) = 6 \times \beta(a, b, i))$
- $\beta(a, b, i + 1) = 6 \times \beta(a, b, i)$ надо заменить на $\exists x, y(x = \beta(a, b, i + 1)) \wedge (y = \beta(a, b, i)) \rightarrow (x = 6 \times y)$
- Наша ближайшая цель доказать, что график вычислимой функции арифметичен.

Задачи

1. Докажите, что существует машина Тьюринга, которая принимает квадрат своего номера, а все остальные входы отвергает.
2. Докажите, что существует два различных алгоритма A и B, что A печатает номер B, а B печатает номер A.
3. Докажите, что предикат p – это n-ое простое число арифметичен.
4. Обозначим $K(x)$ минимальный номер машины Тьюринга, которая на пустом входе печатает x и останавливается. Докажите, что функция $K(x)$ не является вычислимой.
5. Существует ли алгоритм, проверяющий, работает ли данная машина Тьюринга полиномиальное время?