Московский государственный технический университет
имени Н. Э. Баумана
Кафедра «Программное обеспечение ЭВМ и информационные технологии»

лекции

Логика и теория алгоритмов

Алексей Иванович Белоусов

Вёрстка: Р. И. Инфлянскас
Иллюстрации: А. С. Никичкин

15 февраля 2013 г.

Организационные вопросы

Форма сдачи: зачёт

Шкала оценок:

Модули	$3 \cdot 30 = 90$
Прилежание	10

Аудитория: 226л

Литература

1. Мендельсон. Введение в математическую логику.
2. Непейвода. Прикладная логика.
3. Катленд. Вычислимость.

Содержание

1. Теория алгоритмов

Предтечи

Парадокс Рассела Пусть множество Y определяется следующим образом:

$$Y = \{X : |X| \geqslant 3\}$$

Это множество содержит, к примеру, множества

$$X_1 = \{a, b, c\}$$
$$X_2 = \{a, b, c, d\}$$
$$X_3 = \{a, b, c, d, e\}$$

Но тогда оно само имеет как минимум 3 элемента, а значит: $Y \in Y$.
Гилберт предложил следующее:

$Z = \{X : X \notin X\}$
$Z \in Z \Rightarrow Z \notin Z$
$Z \notin Z \Rightarrow Z \in Z$
$Z \notin Z \Leftrightarrow Z \in Z$
$Z \notin Z \Rightarrow Z \neq Z$, то есть $Z \in Z \Rightarrow Z \notin Z \Rightarrow (Z \in Z)\&(Z \notin Z)$ — *противоречие!*

Самоприменимые прилагательные: Самоприменимые прилагательные — прилагательные, которые описывают сами себя.

1. Трёх-слож-ный — три слога, слово описывает само себя.
2. Несамоприменимый — *противоречие*!

Теорема Гёделя Гёдель показал, что в теории могут быть утверждения, которые нельзя ни доказать, ни опровергнуть.
Чтобы полностью проанализировать математику, надо выйти за её пределы.
Рис. 1.0
Основатель теории алгоритмов — Тьюринг (работал шифровальщиков в I мировую войну). Теория алгоритмов тесно связана с криптографией.

1.1. Понятие алгоритма в интуитивном смысле слова

Входные данные и результат — конструктивные объекты.
Конструктивный объект — слово в конечном алфавите.
Множество X — множество входных слов, Y — множество выходных слов. Причём: $X \subseteq V^*, Y \subseteq W^*$
Рис. 1.1
A — алгоритм типа XY : $\quad A : X \to Y$
A — частичный алгоритм типа XY : $\quad A : X \dot{\to} Y$
Алгоритм определяет частичную функцию, которая в качестве области определения использует подмножество X, а значения — подмножество Y.
Признаки алгоритма:

1. **Признак детерминированности** Алгоритм определяет детерминированный процесс. Детерминированный процесс осуществляется за конечное число шагов, и на каждом шаге однозначно определено продолжение процесса или его прекращение.

2. **Признак массовости** Любой алгоритм может осуществлять преобразования в достаточно широком множестве слов.
3. **Признак результативности** Алгоритм должен через конечное число шагов дать определённый результат.

Словарная (вербальная) функция: $V, W \quad f : V^* \underset{\cdot}{\to} W^*$
$V = W \quad f(x) \rightleftharpoons xx = x^2 \quad f : V^* \to V^*$

Функции идентификации $x \sqsubseteq y \Leftrightarrow (\exists y_1, y_2)(y = y_1 x y_2)$ — слово x входит в слово y.

$$g(y) = \begin{cases} \lambda, \text{ если } x \sqsubseteq y \\ y, \text{ иначе} \end{cases}$$

Вербальная функция называется вычислимой в интуитивном смысле слова, если существует алгоритм $A_f : V^* \to W^*$, что $(\forall x \in V^*)(!A_f(x) \Leftrightarrow x \in D(f))\&(A_f(x) = f(x))$

Пусть $A : V^* \underset{\cdot}{\to} W^*$. Тогда $(x \in V^*)!A(x)$ означает, что алгоритм A применим к слову x. $\neg!A(x)$ — алгоритм A не применим к слову x.

Результат алгоритма: $A(x) \in W^*$

1.2. Машина Тьюринга

Алан Тьюринг — английский математик.

Рис. 1.2

Рис. 1.3

Рис. 1.4

Машина Тьюринга — полубесконечная лента, разделённая на буквы.

$\circledast$ — маркер начала ленты.

$\square$ — символ пробела.

Блок управления может находиться в любом состоянии из множества состояний: $Q = q_0, \dots, q_f$

Запись команды:

$$qa \to rb, \begin{Bmatrix} S \\ L \\ R \end{Bmatrix} \quad q, r \in Q,\ a, b \in V \cup \{\circledast, \square\}$$

означает следующее: если в состоянии q обозреваемый символ a, то перейти в состояние r, записать b и сдвинуться (L — на 1 символ влево, R — на 1 символ вправо, S — остаться на месте).

Входное слово записывается на ленте без всяких пробелов буква за буквой. Первый пробел — конец слова. Потом идёт бесконечное число пробелов.

Когда машина Тьюринга даёт результат, она головка останавливается на маркере начала ленты в заключительном состоянии. Сразу после этого идёт результат, потом — пробелы.

Вместо буквы после состояния может идти параметр, к примеру: $\alpha \in \{a, b, c\}$ — любой из символов $\{a, b, c\}$.

Пример 1. Что делает машина Тьюринга со следующей системой команд?

$$
\begin{array}{cc}
q_0\circledast \to q_0\circledast, R & q_2 b \to q_2 b, L \\
q_0 a \to q_0 a, R & q_2\circledast \to q_f\circledast, L \\
q_0 b \to q_0 b, R & q_1\square \to q_3\square, L \\
q_0 c \to q_0 c, R & q_3 a \to q_3\square, L \\
q_1 a \to q_1 a, R & q_3 b \to q_3\square, L \\
q_1 b \to q_1 b, R & q_3 c \to q_3\square, L \\
q_1 c \to q_1 c, R & q_3\circledast \to q_3\square, L \\
q_0\square \to q_2\square, L & q_3\circledast \to q_f\circledast, S \\
q_2 a \to q_2 a, L &
\end{array}
$$

Ответ: стирает слова, которые содержат букву c.

Формально машина Тьюринга определяется как следующий кортеж:

$$T = (V, Q, q_0, q_f, \circledast, \Box, S, L, R, \delta),$$

где δ — система команд:

$$\delta : Q \times V' \to 2^{Q\times V'\times\{S,L,R\}}, \text{ где } V' = V \cup \{\circledast, \Box\}$$
$$\delta : Q \times V' \dot{\to} Q \times V' \times \{S, L, R\}$$

В дальнейшем мы будем иметь дело только с детерминированными машинами Тьюринга.

Определение 1 Конфигурация.

$$C = (q, x, ay) \in Q \times V'^* \times V'V'^*, \text{ то есть } q \in Q,\ x, y \in V'^*,\ a \in V'$$

a — символ, обозреваемый головкой, y — цепочка, стоящая сразу после a (с точностью до любой цепочки пробелов, стоящих в конце).

В любой машине выделяется начальная конфигурация:

$$C_0 = (q_0, \lambda, \circledast x \Box)$$

и конечная конфигурация:

$$C_f = (q_f, \lambda, \circledast y \Box)$$

Определение 2 Отношение непосредственной выводимости.

$$C = (q, x, ay) \vdash_{\mathcal{T}} \begin{cases} (r, x, by), & \text{если в системе команд } \delta \text{ есть команда } qa \to rb, S \\ (r, x', cby), & \text{если в системе команд } \delta \text{ есть команда } qa \to rb, L\,(x = x',\ c \neq \lambda) \\ (r, xb, dy'), & \text{если в системе команд } \delta \text{ есть команда } qa \to rb, R\,(dy' = y) \end{cases}$$

Рис. 1.5.

Определение 3 Выводимость на множестве конечных конфигураций машины Тьюринга.

$$C_1, C_2, \ldots, C_n, \ldots\ n \geqslant 1$$
$$(\forall i \geqslant 1)(C_i \vdash_{\mathcal{T}} C_{i+1}), \text{ если } C_{i+1} \text{ определена в последовательности}$$
$$C_1 \vdash_{\mathcal{T}} C_2 \vdash_{\mathcal{T}} \ldots \vdash_{\mathcal{T}} C_n - \text{длина } = n - 1 (n \geqslant 1)$$
$$C \vdash^*_{\mathcal{T}} C' \rightleftharpoons \text{ существует вывод } C_1 \vdash_{\mathcal{T}} C_2 \vdash_{\mathcal{T}} \ldots \vdash_{\mathcal{T}} C_n = C', n \geqslant 1)$$

В детерминированной машине Тьюринга из каждой конфигурации

Пусть дана машина Тьюринга $\mathcal{T}$ и слово $x \in V^*$

$$!\mathcal{T}(x) \rightleftharpoons (q_0, \lambda, \circledast x \Box) \vdash^*_{\mathcal{T}} (q_f, \lambda, \circledast y \Box)\ y \rightleftharpoons \mathcal{T}(x)$$

Не успел перепечатать следующее выражение!

$$\neg!\mathcal{T}(x) \rightleftharpoons [(q_0, \lambda, \ldots$$

Определение 4. Вербальная функция

$$f : V^* \dot{\to} V^*$$

вычислима по Тьюрингу, если может быть построена $\mathcal{T}$_f с рабочим алфавитом $V_1 \supseteq V (\forall x \in V^*)(!\mathcal{T}_f(x) \Leftrightarrow x \in D(f))\&(\mathcal{T}_f(x) = f(x))$

Теорема 1 Тезис Тьюринга. *Всякая функция, вычислимая в интуитивном смысле слова вычислима по Тьюрингу.*

Пример 2 Машина Тьюринга, стирающая всё, если есть вхождение слова.

$$\mathcal{T}_1 : \mathcal{T}_1(x) = \begin{cases} \lambda, \text{ если } aab \sqsubseteq x \\ x \text{ иначе} \end{cases}, \text{ где } V = \{a, b\}$$

$$\begin{aligned}
q_0 \circledast &\to q_0 \circledast, R \\
q_0 a &\to q_1 a_1, R \\
q_0 b &\to q_0 b_1, R \\
q_1 a &\to q_2 a_1, R \\
q_1 b &\to q_0 b, R \\
q_2 a &\to q_2 a, R \\
q_2 b &\to q_3 b, R \\
q_3 \alpha &\to q_3 \alpha, R, \alpha \in \{a, b\} \\
q_3 \Box &\to q_4 \Box, L \\
q_4 \alpha &\to q_4 \Box, L, \alpha \in \{a, b\} \\
q_4 \circledast &\to q_f \circledast, S \\
q_0 \Box &\to q_5 \Box, L \quad q_5 \text{ — движение в случае неуспешного поиска} \\
q_1 \Box &\to q_5 \Box, L \\
q_2 \Box &\to q_5 \Box, L \\
q_5 \alpha &\to q_5 \alpha, L, \alpha \in \{a, b\} \\
q_5 \circledast &\to q_5 \circledast, S
\end{aligned}$$

Прогонка:

$$(q_0, \lambda, \circledast aaaabab\Box) \vdash (q_0, \circledast, aaaabab\Box) \vdash (q_1, \circledast a, aaabab\Box) \vdash (q_2, \circledast aa, aabab\Box) \vdash$$
$$\vdash (q_2, \circledast aaa, abab\Box) \vdash (q_2, \circledast aaaa, bab\Box) \vdash (q_3, \circledast aaaab, ab\Box) \vdash^2 (q_3, \circledast aaaabab, \Box\Box) \vdash$$
$$\vdash (q_4, \circledast aaabab, b\Box\Box) \vdash^6 (q_4, \circledast, \Box) \vdash (q_4, \lambda, \circledast\Box) \vdash (q_f, \lambda, \circledast\Box)$$

Пример 3.

$$\mathcal{T}_2 : (q_0, \lambda, \circledast x\Box) \vdash^* (q_f, \lambda, \circledast \# x\Box), \quad V = V_0 \cup \{\#\}, \# \notin V_0,\ x \in V_0^*$$

$$\begin{aligned}
q_0 \circledast &\to q_0 \circledast, R \\
q_0 \Box &\to q_f \#, L \\
q_0 \alpha &\to q_\alpha \#, R \quad \alpha \in V_0 \\
q_\alpha \beta &\to q_\beta \#, R \quad \alpha, \beta \in V_0 \\
q_\alpha \Box &\to q_1 \alpha, L \\
q_1 \gamma &\to q_1 \gamma, L \quad \beta \in V_0 \cup \{\#\} \\
q_1 \circledast &\to q_f \circledast, S
\end{aligned}$$

Прогонка:

$$V_0 = \{a, b\} \quad (q_0, \lambda, \circledast ab\Box) \vdash (q_0, \circledast, ab\Box) \vdash (q_a, \circledast \#, b\Box) \vdash (q_b, \circledast \# a, \Box) \vdash (q_1, \circledast \#, ab\Box) \vdash$$
$$\vdash (q_1, \circledast, \# ab\Box) \vdash (q_1, \lambda, \circledast \# ab\Box) \vdash (q_f, \lambda, \circledast \# ab\Box)$$

Пример 4.

$$\mathcal{T}_3 : (q_0, \lambda, ⊛\#x\square) \vdash^* (q_f, \lambda, ⊛x\square),\text{ где } x \in V_0^*,\ V = V_0 \cup \{\#\}$$

$$\begin{aligned}
q_0⊛ &\to q_0⊛, R\\
q_0\# &\to q_\#\#, R\\
q_\#\alpha &\to q_\alpha\#, L\\
q_\alpha\# &\to q_0\alpha, R\\
q_\#\square &\to q_1\square, L\\
q_1\alpha &\to q_1\alpha, L \quad \alpha \in V_0\\
q_1⊛ &\to q_f⊛, S\\
q_1\# &\to q_1\square, L
\end{aligned}$$

Прогонка:

$(q_0, \lambda, ⊛\#abc\square) \vdash (q_0, ⊛, \#abc\square) \vdash (q_\#, ⊛, \#, abc\square) \vdash (q_a, ⊛, \#\#bc\square) \vdash (q_o, ⊛a, \#bc\square) \vdash$
$\vdash (q_\#, ⊛a\#, bc\square) \vdash (q_b, ⊛a, \#\#c\square) \vdash (q_0, ⊛ab, \#c\square) \vdash (q_\#, ⊛ab\#, c\square) \vdash (q_c, ⊛ab, \#\#\square) \vdash$
$\vdash (q_0, ⊛abc, \#\square) \vdash (q_\#, ⊛abc\#, \square) \vdash (q_1, ⊛abc, \#\square) \vdash (q_1, ⊛ab, c\square\square) \vdash^3 (q_1, \lambda, ⊛abc\square) \vdash$
$\vdash (q_f, \lambda, ⊛abc\square)$

Пример 5. Пусть требуется сдвинуть на заранее заданное количество символов влево.
Вместо:

$$\begin{aligned}
q_1\alpha &\to q_1\alpha, L \quad \alpha \in V_0\\
q_1⊛ &\to q_f⊛, S\\
q_1\# &\to q_1\square, L
\end{aligned}$$

из предыдущего примера вставим:

$$\begin{aligned}
q_1\#\square &\to q_2\square, L\\
q_2\alpha &\to \alpha, L \quad (\alpha \in V_0)\\
q_2\# &\to q_\#\#, R\\
q_\#\# &\to q_\#\#, R\\
q_2⊛ &\to q_f⊛, S
\end{aligned}$$

Любая модель алгоритмов должна иметь:

1. Описание модели.
2. Понятие эквивалентных алгоритмов.
3. Способы сочетания алгоритмов.
4. Универсальный алгоритм.
5. Понятие разрешимого и перечислимого множества.
6. Алгоритмически неразрешимых проблем.

1.3. Нормальные алгорифмы Маркова

Определение 5 Вхождение слова.

$$V,\ x,y \in V^* \quad x \sqsubseteq y \rightleftharpoons (\exists y_1, y_2)(y = \underbrace{y_1}_{\text{левое крыло}} \underbrace{x}_{\text{основа}} \underbrace{y_2}_{\text{правое крыло}})$$

$$(\forall x)(\lambda \sqsubseteq x) \& (x \sqsubseteq x)$$

$$x \sqsubseteq y, y \sqsubseteq z \Rightarrow x \sqsubseteq z \quad (y_1, x, y_2), \text{ где } y = y_1 x y_2 \quad y_1 \star x \star y_2,\ \star \notin V$$

Самое левое вхождение слова: $\star \star x$.
Первым вхождением слова x в слово y которое имеет наименьшую длину правого крыла.

Определение 6 Формула подстановки.

$$\omega : u \to v, \quad u, v \in V^*, \to \notin V$$

Определение 7 Применимость. Если $x, u \sqsubseteq x$, то говорят, что ω применима к x (подходит для слова x).

Первое вхождение u в x: $x_1 \star u \star x_2$.

$$y \rightleftharpoons x_1 v x_2 \rightleftharpoons \omega x$$

Например, $x =$ входит, $\omega :$ вход $\to$ уход. $\omega x =$ уходит.

Определение 8 Нормальный алгорифм.

$$\mathcal{A} = (V, \mathcal{S}, \mathcal{P})$$

$\mathcal{A}$ — алгоритм, $\mathcal{P}$ — заключительные формулы.

Схема нормального алгорифма:

$$\begin{cases} u_1 \to [\cdot] v_1 \\ u_2 \to [\cdot] v_2 \\ \vdots \\ u_n \to [\cdot] v_n \end{cases}$$

Пример 6. Пусть дана схема:

$$\mathcal{A}_0 : \begin{cases} \#a & \to a\# \quad V = \{a, b, \#\} \\ \#b & \to b\# \\ \ \# & \to \cdot aba \\ & \to \# \end{cases}$$

$$\mathcal{A}_0 : x = aba \vdash \#aba \vdash a\#ba \vdash ab\#a \vdash aba\# \vdash\cdot abaaba$$

$$\mathcal{A}_0 : x = x(1)x(2)\dots x(k) \vdash \#x(1)x(2)\dots x(k) \vdash x(1)\#x(2)\dots x(k) \vdash$$
$$\vdash x(1)x(2)\#\dots x(k) \vdash \dots \vdash x(1)x(2)\dots x(k)\# \vdash\cdot x(1)x(2)\dots x(k)aba \quad (k \geqslant 1)$$

Пример 7.

$$\mathcal{A}_1 : \left\{ \to \cdot aba (\forall x \in a, b^*)(\mathcal{A}_1 : x \vdash\cdot abax) \right.$$

$$\mathcal{A} = (V, \mathcal{S}, \mathcal{P}), -$$

алгорифм $\mathcal{A}$ просто непосредственно переводит x в y. $x \vdash y \rightleftharpoons y = \omega x$, где ω — первая входящая в S подходящая для x формула, не являющаяся заключительной.

Алгорифм $\mathcal{A}$ непосредственно заключительно переводит x в y: $A : x \vdash\cdot\, y$

Алгорифм $\mathcal{A}$ просто переводит x в y: $\mathcal{A} : x \vDash y \rightleftharpoons$ Существует последовательность слов

$$x_0 = x, x_1, x_2, \dots, x_n = y, \text{ где } (\forall i = \overline{0, n-1})(\mathcal{A} : x_i \vdash x_{i+1})$$
$$\mathcal{A} : x \vDash y \rightleftharpoons (\mathcal{A} : \ \vdash\cdot\, y) \rightleftharpoons (\mathcal{A} : x \vdash\cdot\, y) \vee (\exists z)(\mathcal{A} : x \vDash z \vdash\cdot\, y)$$

Определение 9 Процесс работы нормального алгорифма со словом x. Это конечная или бесконечная последовательность слов:

$$x_0 = x, x_1, \dots, x_n, \dots \text{ такая, что } (\forall i \geqslant 0)(\mathcal{A} : x_i \vdash x_{i+1}) \vee (\mathcal{A} : x_i \vdash\cdot\, x_{i+1}),$$
если x_{i+1} определено в последовательности.

При этом слово x_n не определено тогда и только тогда (по определению):

1. $n - 1 = 0$ и $\mathcal{A} : \neg x_0 = x$
2. $n > 0$ т. е. x_{n-1} определено, но $\mathcal{A} : \neg x_{n-1}$
3. $\mathcal{A} : x_{n-2} \vdash\cdot\, x_{n-1}, n \geqslant 2$

Пусть $x = x_0, x_1, \dots, x_n$ — процесс работы $\mathcal{A}$ с x (является конечным). Тогда x_n называется процессом работы нормального алгорифма $\mathcal{A}$ со словом x и обозначается $\mathcal{A}(x)$.

$!\mathcal{A}(x)$ — алгорифм $\mathcal{A}$ применим к слову x.
$\neg!\mathcal{A}(x)$ — алгорифм $\mathcal{A}$ не применим к слову x.

Определение 10.

$$f : V^* \underset{\cdot}{\rightarrow} V^*$$

называется вычислимой по Маркову, если может может быть построен нормальный алгорифм $\mathcal{A}_f$ в алфавите V такой, что

$$(\forall x \in V^*)(!\mathcal{A}_f(x) \Leftrightarrow x \in D(f))\&(\mathcal{A}_f(x) = f(x))$$

Теорема 2 Принцип нормализации. *Любая вербальная функция, вычислимая в интуитивном смысле слова, вычислима по Маркову.*

$$V = \{a_1, \dots, a_n\}, \# \notin V$$

$$R_c : \begin{cases} \#\xi \to \xi\# & \xi \in V) \\ \# \to \cdot x_0 & x_0 \text{ — произвольное фиксированное слово в } V \qquad \text{правое слово} \\ \underset{\cdot}{\rightarrow} \# & \end{cases}$$

$$V, \alpha, \beta \notin V$$

$$\mathcal{A}_2 : \begin{cases} \alpha\xi & \to \xi\beta\xi\alpha \quad (\xi \in V) \\ \beta\xi\eta & \to \eta\beta\xi\alpha \quad (\eta, \xi \in V) \\ \beta & \to \\ \alpha & \to \cdot \\ & \to \alpha \end{cases}$$

$$\lambda \vdash \alpha \vdash\cdot\, \alpha, a \in V$$
$$a \vdash \alpha a \vdash a\beta a\alpha \vdash aa\alpha \vdash\cdot\, aa$$