Логика и алгоритмы, лекция 22

лектор: Кудинов Андрей Валерьевич

11 мая 2021 г.

План

План лекции:

- Неформальное представление об алгоритмах.
- Вычислимые функции
- Вычислительные модели
- Тезис Чёрча–Тьюринга
- Машины Тьюринга

Неформальное представление об алгоритмах.

- Алгоритм есть строго определенное конечное предписание выполнить некоторую последовательность действий (может быть бесконечную).
- Для данного алгоритма $\mathcal{A}$ определены:
  - область возможных исходных данных X;
  - область возможных значений Y.

  В качестве данных обычно рассматриваются слова $X = \Sigma^*$, где Σ — конечный алфавит, или числа $X = \mathbb{N}^n$.

Свойства алгоритма

- Процесс применения алгоритма $\mathcal{A}$ к данным $x \in X$ происходит по шагам.
- Процесс или заканчивается после конечного числа шагов с результатом $y \in Y$, или останавливается без результата или продолжается бесконечно.
- Таким образом, с алгоритмом $\mathcal{A}$ связывается частичная функция $f : X \to Y$.
  Мы будем говорить:

  «Алгоритм $\mathcal{A}$ вычисляет функцию f.»

Частичные функции

Определение

Частичной функцией $f : X \to Y$ называется подмножество $f \subseteq X \times Y$ такое, что из $\langle x, y_1 \rangle \in f$ и $\langle x, y_2 \rangle \in f$ следует $y_1 = y_2$.

Пишем $f(x) = y$ вместо $\langle x, y \rangle \in f$;

$!f(x)$ вместо $\exists y\, f(x) = y$.

Областью определения частичной функции f называется множество $dom(f) := \{x \in X : \exists y \in Y \ \langle x, y \rangle \in f\}$.

Областью значений частичной функции f называется множество $rng(f) := \{y \in Y : \exists x \in X \ \langle x, y \rangle \in f\}$.

Вычислимые функции

Частичная функция $f : X \to Y$ **вычислима**, если она вычисляется некоторым алгоритмом.

В частности, можно говорить о вычислимых функциях $f : \Sigma^* \to \Sigma^*$, $f : \mathbb{N}^k \to \mathbb{N}$ и т.д.

на входе $x \in X$

A заканчивает работу, если $\exists!\, f(x)$ и выдает $f(x)$

A незакан.. раб. (зацикливается), если $f(x)$ неопр. $x \notin dom(f)$

Вычислительные модели

- Машины Тьюринга (А. Тьюринг, Э. Пост)
- Частично рекурсивные функции (К. Гёдель, С. Клини)
- Лямбда-исчисление (А. Чёрч)
- Алгорифмы Маркова
- Машины с неограниченными регистрами
- Pascal, C, Java, Lisp, Python, ...

Эквивалентность вычислительных моделей

Теорема

Каждая из вышеперечисленных моделей определяет один и тот же класс вычислимых частичных функций $f : \Sigma^* \to \Sigma^*$.

Такие модели (языки программирования) называются полными по Тьюрингу.

Тезис Чёрча–Тьюринга

Тезис

Любая вычислимая в интуитивном смысле частичная функция $f : \Sigma^* \to \Sigma^*$ вычислима на машине Тьюринга.

Замечание

Это утверждение не является математическим, но говорит об адекватности математической модели (вычислимости по Тьюрингу) реальному явлению (вычислимости).

Все попытки построения более общих вычислительных моделей неизбежно приводили к тому же самому классу вычислимых функций.

Физический тезис Чёрча–Тьюринга

Текущему уровню знаний не противоречит и более сильный

Тезис

Всякая функция $f : \Sigma^* \to \Sigma^*$, вычислимая на (идеализированном) физически реализуемом устройстве, вычислима на машине Тьюринга.

Замечание

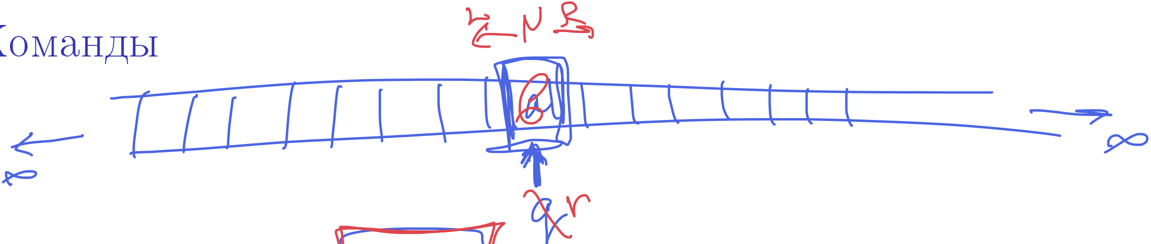
Физический тезис предполагает возможность аналогового вычисления, квантово–механические эффекты и т.д.

Машины Тьюринга

Машина Тьюринга задаётся конечными

- рабочим алфавитом Σ, содержащим символ $\#$ (пробел);
- множеством состояний Q, содержащим состояния q_1 (начальное) и q_0 (конечное);
- набором команд (программой) P.

Команды



- Команды имеют вид $\boxed{qa \to rb\nu}$, где $q, r \in Q$, $a, b \in \Sigma$ и $\nu \in \{L, N, R\}$.
  «прочтя символ a в состоянии q перейти в состояние r, заменить содержимое ячейки на b и сместиться влево (L), остаться на месте (N) или сместиться вправо (R) на одну ячейку, в зависимости от значения ν»

$q_1 \# \to q_2 0 R$

- Требуется, чтобы в программе P была ровно одна команда с левой частью qa для каждого $q \in Q \setminus \{q_0\}$ и $a \in \Sigma$.

Соглашение: команды вида $qa \to qaN$, приводящие к зацикливанию, можно не указывать.

Машина Тьюринга есть набор $M = \langle Q, \Sigma, P, q_0, q_1 \rangle$.

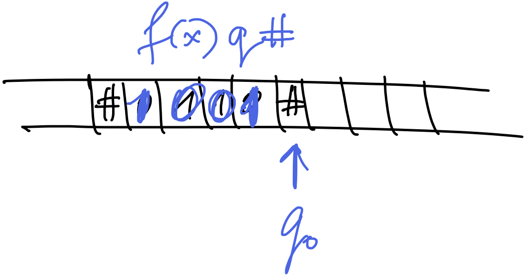
Пример машины Тьюринга

Пусть $\Sigma = \{\#, 0, 1\}$, $Q = \{q_0, q_1\}$, а P состоит из следующих команд:

$$\begin{array}{lcl} q_1\# & \mapsto & q_1\#R \\ q_1 0 & \mapsto & q_1 1R \\ q_1 1 & \mapsto & q_1 0R \end{array}$$

Что делает эта машина Тьюринга?

Модифицируем программу.



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

Пример машины Тьюринга

Пусть $\Sigma = \{\#, 0, 1\}$, $Q = \{q_0, q_1, q_2\}$, а P состоит из следующих команд:

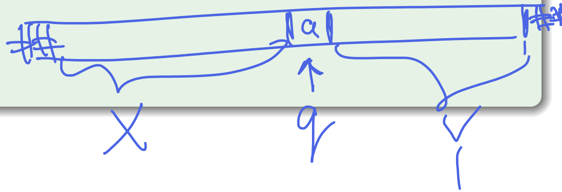
$$
\begin{array}{lcl}
q_1\# & \mapsto & q_1\#R \\
q_1 0 & \mapsto & q_2 1R \\
q_1 1 & \mapsto & q_2 0R \\
q_2 0 & \mapsto & q_2 1R \\
q_2 1 & \mapsto & q_2 0R \\
q_2\# & \mapsto & q_0\#N
\end{array}
$$

Конфигурации

Предположение: лента содержит лишь конечное число символов, отличных от $\#$.

Конфигурация машины M определяется содержимым ленты, состоянием и положением головки. Конфигурация записывается словом вида $XqaY$, где

- $XaY \in \Sigma^*$ есть содержимое ленты,
- $q \in Q$ есть состояние M,
- головка обозревает символ a.



Функция, вычислимая машиной Тьюринга

$\Sigma \setminus \Delta = \{\#, \ldots\}$

Пусть $\Delta \subset \Sigma$ и $\# \notin \Delta$.

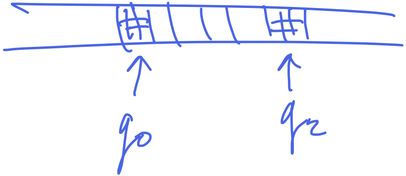
M **вычисляет** частичную функцию $f : \Delta^* \to \Delta^*$, если для каждого $x \in \Delta^*$

- если $x \in dom(f)$, то начав работу в конфигурации $q_1\#x$, машина M останавливается в конфигурации $q_0\#f(x)$;
- если $x \notin dom(f)$, то машина M не останавливается. в конфиг. $q_1\#x$

f - вычислима, если $\exists M$ - маш. Тьюр., кот. её вычисляет.

Машина M из примера (почти) вычисляет функцию $\mathrm{neg} : \{0,1\}^* \to \{0,1\}^*$, заменяющую в данном слове 0 на 1 и 1 на 0. Чтобы вернуть головку в начало модифицируем M:

$$
\begin{array}{rcl}
q_1\# & \mapsto & q_1\#R \\
q_10 & \mapsto & q_21R \\
q_11 & \mapsto & q_20R \\
q_20 & \mapsto & q_21R \\
q_21 & \mapsto & q_20R \\
q_2\# & \mapsto & q_3\#L \\
q_30 & \mapsto & q_30L \\
q_31 & \mapsto & q_31L \\
q_3\# & \mapsto & q_0\#N
\end{array}
$$



Упражнения

Построить машины Тьюринга, вычисляющие следующие функции над алфавитом $\{0, 1\}$:

- $f(x) = xx$ (копирование слова)
- $g(x_1 \dots x_n) = \sum_{i=1}^{n} x_i \mod 2$
  (сумма битов по модулю 2)