Листок 4

Абстрактные вычислительные машины, рекурсия

Алгоритм — конечный набор каких-либо инструкций, позволяющий по входным данным получать однозначный результат. Понятно, что это не математическое определение, однако оно полезно для общего понимания понятия алгоритма. Если для некоторой функции существует алгоритм, реализующий её, функция называется вычислимой.

Для более формального, подходящего для теоретического изучения понятия алгоритма удобно вводить абстрактные конструкции, так или иначе отражающие основные свойства вычислимых «в реальном мире» функций.

Машина Тьюринга.

Неформально: машина Тьюринга (МТ) состоит из бесконечной, разбитой на клетки, ленты и головки, представляющей собой автомат. Лента используется для хранения информации, в каждой клетке может быть записан символ или ничего не записано. Для удобства будем пустое содержимое клетки обозначать специальным символом Λ (и отразим это в формальном определении). Головка — активная часть МТ, на каждом шаге она размещается над одной из клеток ленты и видит её содержимое. Кроме того, на каждом шаге головка находится в одном из состояний. На каждом шаге МТ делает три элементарных действия в зависимости от состояния головки и видимого символа:

— в клетку под головкой записывается некоторый символ,

— головка сдвигается влево или вправо на одну клетку, либо остаётся на месте,

— состояние головки меняется на некоторое состояние.

Конкретный способ изменения состояний, букв на ленте и движения головки определяется программой МТ. МТ работает тактами, которые выполняются один за другим. Программа определяет переход $\langle S, q\rangle \to \langle S', q', \{L, R, N\}\rangle$, здесь S, S' — символы на ленте до и после такта МТ, q, q' — состояния головки до и после такта, L, R, N — команды «сдвинуться влево», «сдвинуться вправо» и «не двигаться» соответственно.

До выполнения программы на ленту написано входное слово (по одной букве в клетке) и, если не указано обратное, головка МТ находится над первой буквой входного слова в состоянии q_0. После этого начинается выполнение программы. В результате МТ должна совершить необходимые действия (подразумеваемые конкретной задачей). Для останова МТ вводятся специальные состояния останова, попав в которые головка останавливается, а МТ прекращает работу.

Формально: МТ есть набор $\langle Q, \Sigma, \Gamma, \Lambda, q_0, Q_f, \delta\rangle$, где

— Q есть конечное множество, называемое множеством состояний,

— Σ есть конечное множество, называемое алфавитом МТ,

— $\Gamma \subset \Sigma$ есть конечное множество, называемое входным алфавитом,

— $\Lambda \in \Sigma$ есть пустой символ (только он может быть написан на ленте бесконечное число раз),

— $q_0 \in Q$ есть начальное состояние,

— $Q_f \subset Q$ есть множество конечных состояний, чаще всего мы будем считать, что конечные состояния есть q_{accept} и q_{reject} – остановившись в первом, МТ принимает слово, во втором – отвергает,

— $\delta : (Q \setminus Q_f) \times \Sigma \to Q \times \Sigma \times \{L, R\}$ есть функция переходов.

Тезис Тьюринга: для любой вычислимой функции существует МТ, вычисляющая эту функцию. Т.е, МТ может реализовать любой алгоритм.

В силу тезиса Тьюринга примем следующие определения рекурсивного и рекурсивно перечислимого языков.

Язык L называется **рекурсивным** или **разрешимым** и пишут $L \in \mathcal{R}$, если для него существует машина Тьюринга, которая останавливается на любом слове и принимает его тогда и только тогда, когда это слово принадлежит языку.

Язык L называется **рекурсивно перечислимым** или **полуразрешимым** и пишут $L \in \mathcal{RE}$, если для него существует машина Тьюринга, которая остановится и примет любое слово из языка, но остановится и отвергнет или не остановится вообще для любого слова не из языка. Альтернативно: язык называется **рекурсивно перечислимым**, если для него существует машина Тьюринга, которая перечисляет все слова этого языка.

Сводимость:

Пусть A и B два языка. Будем говорить, что A (многозначно) сводится к B, и писать $A \leqslant_m B$, если существует вычислимая функция $f : \Sigma^* \to \Sigma^*$, что $\forall \omega \in \Sigma^*$ выполняется $\omega \in A$ тогда и только тогда, когда $f(\omega) \in B$.

Если $A \leqslant_m B$, то

$B \in \mathcal{R} \Rightarrow A \in \mathcal{R}$

$B \in \mathcal{RE} \Rightarrow A \in \mathcal{RE}$

$A \notin \mathcal{R} \Rightarrow B \notin \mathcal{R}$

$A \notin \mathcal{RE} \Rightarrow B \notin \mathcal{RE}$

Для любого множества языков $\mathcal{C}$ определим множество языков $co\mathcal{C}$ следующим образом: для любого языка L верно, что $L \in co\mathcal{C}$ тогда и только тогда, когда $\Sigma^* \setminus L \in \mathcal{C}$.

В соответствии с этим класс co$\mathcal{RE}$ определяется следующим образом:

$$L \in \mathrm{co}\mathcal{RE} \Longleftrightarrow \Sigma^* \setminus L \in \mathcal{RE}$$

4.1: Построить МТ

а) Разрешающую язык $a^n b^n$, $n \in \mathbb{N}_0$; $\Gamma = \{a, b\}$

а) Складывающую два числа в двоичной системе счисления – на вход подаётся два числа, разделённые знаком +, $\Gamma = \{0, 1, +\}$, по окончании работы МТ на ленте должна быть записана сумма в двоичной системе счисления

4.2: Проверить языки на принадлежность $\mathcal{R}$, $\mathcal{RE}$ и co$\mathcal{RE}$. Здесь предполагается, что для МТ M её описание есть $\langle M \rangle$

1) $\mathsf{SelfAccept} = \{\langle M \rangle | M$ принимает $\langle M \rangle\}$

2) $\mathsf{SelfReject} = \{\langle M \rangle | M$ отвергает $\langle M \rangle\}$

3) $\mathsf{SelfHalt} = \{\langle M \rangle | M$ останавливается на $\langle M \rangle\}$

4) $\mathsf{SelfNotHalt} = \{\langle M \rangle | M$ не останавливается на $\langle M \rangle\}$

5) $\mathsf{Accept} = \{(\langle M \rangle, \omega) \,| M$ принимает $\omega\}$

6) $\mathsf{Reject} = \{(\langle M \rangle, \omega) \,| M$ отвергает $\omega\}$

7) $\mathsf{Halt} = \{(\langle M \rangle, \omega) \,| M$ останавливается на $\omega\}$

8) $\mathsf{NotHalt} = \{(\langle M \rangle, \omega) \,| M$ не останавливается на $\omega\}$

9) $\mathsf{NeverAccept} = \{\langle M \rangle | M$ не принимает ни одного слова$\}$

10) $\mathsf{NeverReject} = \{\langle M \rangle | M$ не отвергает ни одного слова$\}$

11) $\mathsf{NeverHalt} = \{\langle M \rangle | M$ не останавливается ни на одном слове$\}$

12) $\mathsf{NeverNotHalt} = \{\langle M \rangle | M$ останавливается на каждом слове$\}$

4.3: Проверить языки на принадлежность $\mathcal{R}$, $\mathcal{RE}$ и co$\mathcal{RE}$.

1) $L_1 = \{\langle M \rangle |$ есть вход, на котором M останавливается меньше, чем за $|\langle M \rangle|$ шагов$\}$

2) $L_2 = \{\langle M \rangle |\ |\mathsf{Accept}(M)| \leqslant 2022\}$

3) $L_3 = \{\langle M \rangle |\ |\mathsf{Accept}(M)| \geqslant 2022\}$

4) $L_4 = \{\langle M \rangle |\ \mathsf{Accept}(M)$ конечен$\}$

5) $L_5 = \{\langle M \rangle |\ \mathsf{Accept}(M)$ бесконечен$\}$

6) $L_6 = \{\langle M \rangle |\ \mathsf{Accept}(M)$ счётный$\}$

7) $L_7 = \{\langle M \rangle |\ \mathsf{Accept}(M)$ несчётный$\}$

8) $L_8 = \{\langle M, N \rangle |\ \varepsilon \in \mathsf{Accept}(M) \cup \mathsf{Accept}(N)\}$

9) $L_9 = \{\langle M, N \rangle |\ \varepsilon \in \mathsf{Accept}(M) \cap \mathsf{Accept}(N)\}$

10) $L_{10} = \{\langle M, N \rangle |\ \varepsilon \in \mathsf{Accept}(M) \setminus \mathsf{Accept}(N)\}$

4.4 Проверить языки на принадлежность $\mathcal{R}$, $\mathcal{RE}$ и co$\mathcal{RE}$.

1) $L_{nl} = \{(\langle M\rangle, \omega)\,|M$ никогда не движется влево на входе $\omega\}$

2) $L_{2022l} = \{(\langle M\rangle, \omega)\,|M$ движется влево по крайней мере 2022 раза на входе $\omega\}$

3) $L_x = \{\langle M\rangle|M$ пишет x в какой-то момент при пустом входе$\}$