

Элементы теории алгоритмов

Определение распознавательной задачи:

имеется множество объектов X и определенное подмножество $P \subset X$ (определяемое предикатом $P(x)$), требуется найти эффективную процедуру (т.е. алгоритм), с помощью которой для любого $a \in X$ можно определить $a \in P$ или $a \notin P$ (т.е. высказывание $P(a)$ истинно или ложно), .

При этом распознавательная задача называется *алгоритмически разрешимой* или *алгоритмически неразрешимой* в зависимости от того, имеется или нет алгоритм решения этой задачи.

Конструктивные объекты любого множества X можно *кодировать* словами конечного множества Σ (например, состоящего из двоичных символов 0 и 1) с помощью взаимно-однозначного отображения $f: X \rightarrow \Sigma^*$, где Σ^* - множество всех конечных последовательностей символов алфавита Σ .

Элементы множества Σ^* называются *словами* и подмножества множества Σ^* называются *языками* над алфавитом Σ .

Пример. Кодировка проблема выполнимости (ВЫП)

Формулы алгебры высказываний строятся из следующих элементов.

1. Пропозициональные переменные, принимающие значения 1 (истина) или 0 (ложь).
2. Бинарные операторы \wedge, \vee , обозначающие логические связки И, ИЛИ двух формул.
3. Унарный оператор \neg , который обозначает логическое отрицание.
4. Скобки для группирования операторов и операндов, если необходимо изменить порядок старшинства (приоритетов) операторов, принятый по умолчанию (вначале применяется - \neg , затем \wedge и, наконец, \vee).

Кодировка экземпляров ВЫП

Используется следующий код для алфавита

$$\Sigma = \{X, 0, 1, \wedge, \vee, \neg, (,)\}.$$

1. Символы \wedge, \vee, \neg , и скобки $(,)$ представляют самих себя.
2. Переменная X_i представляется символом X с дописанной к нему последовательностью нулей и единиц — двоичной записью числа i .

Таким образом, алфавит Σ проблемы-языка ВЬП содержит всего восемь символов $X, 0, 1, \wedge, \vee, \neg, (,)$. Все экземпляры ВЬП являются конечными последовательностями символов - словами в этом фиксированном конечном алфавите. Множество кодов всех формул алгебры высказываний образует подмножество $W \subset \Sigma^*$ и множество кодов всех выполнимых формул алгебры высказываний образует подмножество $L \subset W$. Требуется найти эффективную процедуру (т.е. алгоритм), с помощью которой для любого слова $w \in W$ можно определить $w \in L$ или $w \notin L$.

Вывод. Произвольная распознавательная задача универсально формулируется следующим образом:

имеется множество слов $W \subset \Sigma^*$ над некоторым алфавитом Σ и определенный язык $L \subset W$, требуется найти эффективную процедуру (т.е. алгоритм), с помощью которой для любого слова $w \in W$ можно определить $w \in L$ или $w \notin L$.

Интуитивное понятие алгоритма и его математические модели

Под *алгоритмом* интуитивно понимается совокупность инструкций, которые дают решение некоторой массовой задачи.

Общие свойства алгоритма:

- 1) *дискретность алгоритма*;
- 2) *детерминированность алгоритма*;
- 3) *элементарность шагов алгоритма*;
- 4) *массовость алгоритма*.

Так как конструктивные объекты можно кодировать словами конечного алфавита Σ (например, состоящего из двоичных символов

0 и 1), то алгоритм моделируется устройством, перерабатывающим слова алфавита Σ .

Тезис Черча:

класс задач, решаемых в любой формальной модели алгоритма, совпадает с классом задач, которые могут быть решены любым физически реализуемым вычислительным устройством.

Алгоритмически неразрешимые задачи привели к необходимости строго математического определения алгоритма.

Основные варианты математического определения алгоритма - *модели алгоритма:*

- 1) понятие *рекурсивной функции*, введенное Клини в 1936 г.,
- 2) понятие *машины Тьюринга*, введенное Постом и Тьюрингом в 1936 г.,
- 3) понятие *нормального алгорифма*, введенное Марковым в 1954 г.,
- 4) понятие *формальной грамматики*, введенное Хомским в 1957 г.

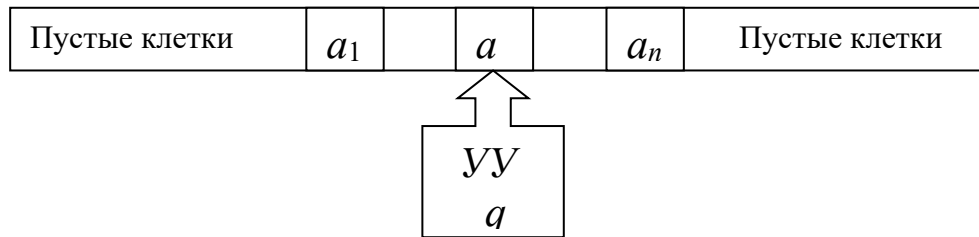
Машины Тьюринга

Реализация модели вычислений с помощью понятия машины Тьюринга.

При построении математической модели алгоритма Пост и Тьюринг исходили из того, что все действия, которые может производить любой алгоритм, можно разложить на некоторые канонические элементарные шаги, выполняемые подходяще устроенными вычислительными машинами.

Такие машины схематически определяются следующим образом.

Схематическое описание работы машины Тьюринга:



- 1) символы внешнего алфавита $\Sigma = \{0, 1, \dots\}$ записываются в ячейки конечной ленты, которая называется *внешней памятью машины*, при необходимости в ячейки записывается символ *, который называется *пустым*
- 2) символы внутреннего алфавита $Q = \{q_S, q_F, \dots\}$ обозначают состояния *управляющего устройства машины (УУ) с просматривающей головкой*, которая может перемещаться вдоль ленты и в каждый момент времени t просматривать одну ячейку,
- 3) *программа машины*

$$\Pi = \{T(q, a) : q \in Q \setminus \{q_F\} \wedge a \in \Sigma\}$$

состоит из команд $T(q, a) = qa \rightarrow q'a'X$, которые в зависимости от состояния машины q и символа a в просматриваемой ячейке УУ заменяют в этой ячейке букву a на букву a' , состояние q на состояние q' и в зависимости от значения $X \in \{R, L, S\}$ сдвигают просматривающую головку либо в соседнюю правую ячейку при $X = R$, либо в соседнюю левую ячейку при $X = L$, либо оставляет головку на месте при $X = S$. При необходимости на ленте достраиваются справа или слева ячейки с пустым символом *,

- 4) машина начинает работать в *начальном* состоянии q_S и заканчивает работать в *заключительном* состоянии q_F .
- 5) Вход машины: слово $w \in \Sigma^*$ на ленте машины T в начальном состоянии q_S .

Выход машины: слово $w' \in \Sigma^*$ на ленте машины T в заключительном состоянии q_F .