

Тема 6. Основы теории алгоритмов

6.1 Интуитивное понятие об алгоритме

Интуитивное понятие алгоритма.

Алгоритм – это правило, сформированное на некотором языке и определяющее процесс переработки допустимых исходных данных в искомые результаты. Допустимыми исходными данными для этого правила являются предложения языка исходных данных.

Правила описания алгоритмов:

- Понятность для исполнителя
 - Массовость (т.е. допустимость для него всех предложений языка исходных данных)
 - Определенность (все шаги алгоритма детерминированы и четко определены)
 - Результативность
-

Алгоритм применим к допустимому исходному данному, если с его помощью, отправляясь от этого исходного данного, можно получить искомый результат. При этом алгоритмический процесс оканчивается после конечного числа шагов и на каждом шаге нет препятствий для его выполнения.

Поскольку требование
завершения алгоритмического
процесса за конечное число
шагов не учитывает реальных
возможностей, связанных с
затратами времени и
ресурсов, то говорят, что при
этом алгоритм потенциально
выполним.

Об источниках алгоритмов:

- Практика;
 - научная теория;
 - совокупность накопленных алгоритмов;
 - изобретательность разработчика.
-

6.2 Три типа алгоритмических моделей

Различный выбор исходных средств формализации приводит к моделям алгоритмов разного вида. Можно выделить три основных типа универсальных алгоритмических моделей, различающихся исходными эвристическими соображениями относительно того, что такое алгоритм.

Первый тип

Связывает понятие алгоритма с вычислениями и числовыми функциями. Наиболее развитая и изученная модель этого типа – рекурсивные функции – первый способ формализации понятия алгоритма.

Второй тип

Основан на представлении об алгоритме как о некотором детерминированном устройстве, способном выполнять в каждый отдельный момент лишь примитивные операции (машина Тьюринга).

Третий тип

Это преобразование слов в произвольных алфавитах, в которых элементарными операциями являются подстановки, т.е. замена куска слова (подслова) другим словом (Нормальный алгоритм Маркова, каноническая система Поста).

Тезис Чёрга

Класс задач, решаемых в любой из этих формальных моделей, и есть класс всех задач, которые могут быть решены интуитивно алгоритмическими методами.

Одна из причин, побудивших заняться теорией алгоритмов, – это подозрительность математиков в связи с накоплением упорно не поддающихся решению задач на нахождение алгоритмов.

Примеры.

- Задача о квадратуре круга.

Требуется найти алгоритм построения с помощью циркуля и линейки квадрата, равновеликого данному кругу.

- Задача трисекции угла.

Требуется найти алгоритм деления произвольного угла с помощью циркуля и линейки на три равные части

Примеры.

- Задача удвоения куба.

Найти алгоритм, позволяющий на стороне любого куба с помощью циркуля и линейки построить сторону куба, объем которого вдвое больше объема заданного куба.

Вторая причина разработки
теории алгоритмов –
необходимость обоснования
математики, поскольку
появления антиномий
привели к тому, что все в
математике стало казаться
неустойчивым.

Пример

Актуально бесконечное множество. Расходуя ограниченное количество ресурсов на каждом шаге, имеющим фиксированную длительность, построить такое множество ни реально, ни потенциально нельзя; нельзя проверить обладает ли все элементы такого множества каким либо свойством из-за ограниченной скорости проверки.

6.3 Кризис теории множеств антиномии. Выводы из антиномий.

До середины XIX века никто не сомневался в истинности математических результатов, залогом чего считалась истинность аксиом. Исследования Лобачевского сокрушили веру в аксиомы и заставили задуматься над тем, что же является фундаментом математики. Оказалось, что все вопросы можно свести к рассмотрению только натуральных чисел и некоторых отношений между ними. Была доказана возможность полной арифметизации матанализа и теории функций

На втором Международном Математическом Конгрессе было заявлено, что теперь в математике остались только целые числа и конечные или бесконечные множества целых чисел. Математика полностью арифметизирована. И вот тогда, когда математика обрела надежный фундамент сама арифметика пошатнулась из-за того что в теории множеств были обнаружены противоречия, вошедшие в историю математики под названием антиномий.

Две теории Кантора из теории МНОЖЕСТВ

Кардинальное число множества M называется его мощностью и обозначается m .

Теорема 1: Для любого кардинально числа m справедливо неравенство $m < 2^m$, где 2^m – мощность множества всех подмножеств бесконечного множества.

Теорема 2: Мощность m' подмножества множеств, имеющих мощность m удовлетворяет неравенству $m' \leq m$

Парадокс Кантора.

Пусть M множество всех множеств обозначим его кардинальное число буквой m . В силу теоремы 1 кардинальное число множества его подмножества 2^m удовлетворяет условию $2^m > m$ с другой стороны множество m есть множество всех множеств его подмножества являются множествами и значит являются элементом M , а их множества являются подмножествами множества M . По теореме 2 должно иметь место неравенство $2^m \leq m$ полученные два неравенства противоречат друг другу

Парадокс Рассела (парадокс брадобреля).

Один из солдат оказался по профессии парикмахером, узнав об этом командир приказал ему брить всех тех и только тех кто сам себя не бреет. Все было хорошо пока не пришло время побрить самого себя оказалось, что побрить самого себя нельзя, так парикмахер брил только тех кто себя не бреет. Не брить себя тоже нельзя, потому что приказано брить всех, кто себя не бреет.

Открытие антиномий потрясло математику и математиков, как землетрясение. Нужно сказать, что математики поразному отреагировали на это явление: одни стали во всем сомневаться, другие на антиномии не отреагировали.

6.4 Машины Тьюринга как модели алгоритмов

В 1937г. английский математик Тьюринг опубликовал работу в которой он уточняя понятие алгоритма, прибегал к воображаемой вычислительной машине.

Машина должна перерабатывать какие-то объекты в исходные результаты. Этими объектами являются слова, построенные из символов-букв.

Машина состоит из бесконечной в обе стороны ленты, разбитой на ячейки, и рабочей головки. Машина работает в дискретные моменты времени $t=0,1,2\dots$. В каждый момент времени во всякой ячейке ленты записана одна буква из некоторого конечного алфавита $A=\{a_0, a_1, a_2\dots a_n\}$, называемым внешним алгоритмом машины, а головка находится в одном состоянии из конечного множества внутренних состояний $Q=\{q_0, q_1\dots q_z\}$. Будем считать, что a_0 -пустой символ, т.е. в ячейке ничего не написано.

Основной частью машины является логический блок, который работает следующим образом. В каждый момент времени рабочая головка обзореваает одну ячейку ленты и в зависимости от того, что там написано и от своего внутреннего состояния она заменяет символ в обзореваемой ячейке, переходит в новое состояние и сдвигает на одну ячейку или остается на месте. Введем для обозначения этого движения символы d_{-1} , d_0 , d_{+1} соответственно.

Т.о. работа МТ задается системой команд вида:

$$q_j * a_i - q_k * a_x * d_y$$

Все случаи сочетания q_j и a_i для разных j и i и все реакции машины на них можно представить в виде функциональной таблицы с двумя входами.

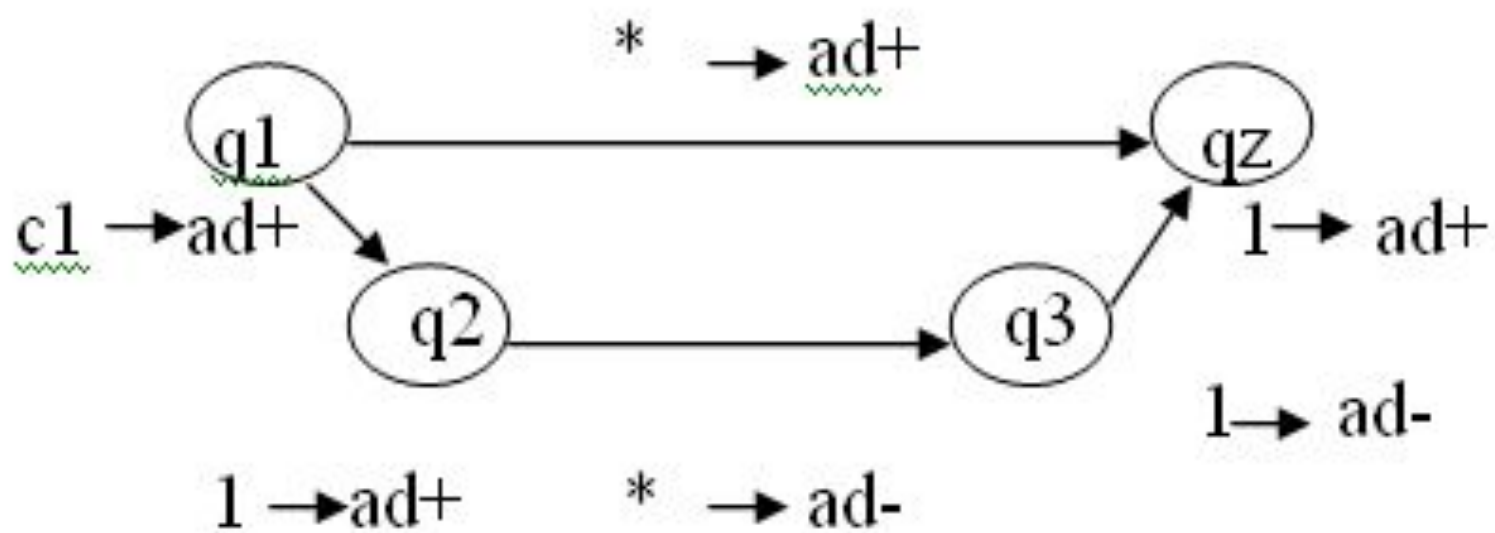
Если головка в состоянии q_j читаем символ a_i , то в следующий момент она записывает в эту ячейку символ a_x . Переход в состояние q_k и в зависимости от того каково d_y головка сдвигает на одну ячейку влево, вправо или остается на месте.

	$q_1, q_2 \dots$	q_j	q_z
a_0			
a_i		$q_k a_x d_y$	
a_n			

Примеры построения машин Тьюринга

1. Построить машину Тьюринга, реализующую «сложение» чисел.

В машине Тьюринга все числа представляются в единичном коде, состоящем из X единиц. Поэтому сложить a и b значит переработать слова $1^a * 1^b$ в слово 1^{a+b} . Это преобразование выполняет машина с 4-мя состояниями и системой команд вида



$q1^* \rightarrow q2 \underline{ad+}$
 $\underline{q1} 1 \rightarrow q2 \underline{ad+}$
 $\underline{q2} 1 \rightarrow q2 1 \underline{ad+}$
 $\underline{qz}^* \rightarrow q3 1 \underline{ad-}$
 $\underline{q3} 1 \rightarrow q3 1 \underline{ad-}$
 $\underline{q3} a \rightarrow qz \underline{ad+}$

Построить машину Тьюринга, которая вычисляет функцию $a(X)=X+1$ число X запишем в виде строки, состоящей из букв 1 (палочек)

	q1	q2
a	1d0q2	-
1	1d-q1	1d0q2 1d0q2

При работе машины возможны два случая:

1. Работа машины после конечного числа шагов прекратили с выдачей сигнала «останов.»
2. Машина никогда не останавливается, ~~никакого результата она не дает.~~

В первом случае говорят, что машина применила к исходному данному, во втором – не применима. Соответствия устанавливаются между теми исходными данными, к которым они применимы, и результат ее работы представляет собой некоторую функцию (в математическом смысле).

Если для функции $f(x)$ можно построить МТ, которая к ней применима, то говорят, что $f(x)$ вычислена по Тьюрингу.

Функцию, для вычисления которой существует МТ, называют ***ВЫЧИСЛИМОЙ***.

Тезис Тьюринга:

Любая вычислимая функция
вычислима по Тьюрингу, или
всякий алгоритм может быть
реализован машиной
Тьюринга.

Проблема остановки:

Можно ли построить машину
То такую что для любой
машины Тк любых исходных
данных а для машины Т.

Ответ на этот вопрос отрица-
тельный, т.к. сформулирована
и доказана следующая
теорема.

Теорема Тьюринга:

Не существует машины T_0 , решающей проблему остановки для произвольной машины T .

Неразрешимость проблемы остановки можно интерпретировать как несуществование общего алгоритма для отладки программы
