МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**МОСКОВСКОЙ ОБЛАСТИ**

**«СЕРГИЕВО-ПОСАДСКИЙ ФИЗИКО-МАТЕМАТИЧЕСКИЙ ЛИЦЕЙ»**

**«Элементы теории алгоритмов. Машина Тьюринга»**

Консультация для учителей

Учитель: Перлова Н..В.

2019-2020 уч.г.

*В 9,10 классах рассматривались определения алгоритма, которые не являются строгими, так как в них используются не определяемые точно термины, например «правило». В 11 классе, для перехода к теме « Элементы теории алгоритмов» сначала вводится уточненное понятие алгоритма.* Определения алгоритма, которые мы с вами рассматривали не являются строгими, так как в них используются не определяемые точно термины, например «правило». Однако математики достаточно долго пользовались интуитивным понятием алгоритма. В рамках подобного определения были сформулированы и успешно применялись на практике алгоритмы для решения таких задач, как нахождение корней квадратного и кубических уравнений, решение систем линейных уравнений (метод Гаусса) и др.

Постепенно математики подходили к постановке и решению все более сложных задач. Так, например, Г. Лейбниц в XVII веке пытался построить общий алгоритм решения любых математических задач. В XX веке эта идея прибрела более конкретную форму: построить алгоритм проверки правильности любой теоремы при любой системе аксиом. Построить такие алгоритмы не удавалось, и математики выдвинули предположение: а вдруг для того или иного класса задач в принципе невозможно построить алгоритм решения. Следовательно, если алгоритма не существует, то они ищут то, чего нет.

На основе этого предположения возникло понятие алгоритмически неразрешимой задачи – задачи, для которой невозможно построить процедуру решения задачи. Надо было научиться математически строго доказывать факт отсутствия соответствующего алгоритма. А это возможно только в том случае, если существует строгое определение алгоритма. Поэтому возникла проблема: построить формальное определение алгоритма, аналогичное известному интуитивному понятию.

Попытки выработать формальное определение алгоритма привели в 20-30 –х годах XX века к возникновению теории алгоритмов. В первой половине XX века разные математики (А. Тьюринг, Э. Пост, А.Н. Колмагоров, А.А.Марков идр.) предложили несколько подходов к формальному определению алгоритма: нормальный алгоритм Маркова, машина Тьюринга, машина Поста и т. д. В дальнейшем было показано, что все эти определения эквивалентны.

Мы рассмотрим формальное определение алгоритма, введенное А. Тьюрингом.

Тьюринг признан одним из основателей информатики и теории искусственного интеллекта, его считают первым теоретиком современного программирования и, наконец, первым в мире хакером. Между прочим, его «хакерская деятельность» внесла во время второй мировой войны существенный вклад в победу союзных войск над германским флотом, а один из коллег Тьюринга однажды сказал: «Я не берусь утверждать, что мы выиграли войну благодаря Тьюрингу. Однако без него могли бы её и проиграть».

Для уточнения понятия алгоритма была предложена абстрактная вычислительная конструкция, которая позже была названа машиной. Тьюринг описал свою машину в 1936 году.

Целью создания такой абстрактной воображаемой машины было получение возможности доказательства существования или не существования алгоритмов решения различных задач. Руководствуясь этой целью, Тьюринг искал как можно более простую, «бедную» алгоритмическую схему, лишь бы она была универсальной.

Прежде чем мы начнем знакомиться с машиной Тьюринга, необходимо сделать замечания относительно объектов, с которыми работают алгоритмы.

1. Замечание. Алгоритм имеет дело не с объектами реального мира, а с некоторыми изображениями этих объектов (объектами работы алгоритмов могут быть только слова).
2. Любой алфавит можно заменить другим (закодировать). Будем считать, что алгоритмы работают со словами, и мы формально описываем объекты – слова, над которыми работают алгоритмы, в некотором алфавите.

**Описание машины Тьюринга**

Машина Тьюринга – это строгое математическое построение, математический аппарат, созданный для решения определённых задач. Этот математический аппарат был назван «машиной» по той причине, что по описанию он похож на вычислительную машину. Принципиальное отличие машины Тьюринга от вычислительной машины заключается в том, что у машины Тьюринга запоминающее устройство есть бесконечная лента: у реальных вычислительных машин устройство может быть сколь угодно большим, но не бесконечным.

В каждой машине Тьюринга есть две части:

1. Неограниченная в обе стороны лента, разделённая на ячейки;
2. Автомат (головка для считывания/записи, управляемая программой)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1**  | **1**  | **1**  | **\***  | **1**  | **1**  |  |  |

^

С каждой машиной Тьюринга связаны два конечных алфавита:

* алфавит входных символов А={а0,а1,а2,…,ам}
* алфавит состояний Q={q0,q1,q2,…,qм}

буква а0 - признак того, что ячейка пуста

состояние q1 – начальное

состояние q0 – пассивное (если машина попала в это состояние, то она закончила свою работу).

Автомат каждый раз видит только одну ячейку. В зависимости от того, какую букву он видит, а так же в зависимости от своего состояния qi, автомат может выполнять следующие действия:

* записать новую букву в обозреваемую ячейку;
* выполнить сдвиг по ленте на одну ячейку вправо/влево или остаться на месте;
* перейти в новое состояние.

Программы для машины Тьюринга представляет собой таблицу, в каждой клетке которой записана команда.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **q0**  | **q1**  | **q2**  | **q3**  | **…**  | **qm**  |
| а0  |  |  |  |  |  |  |
| а1  |  |  |  |  |  |  |
| а2  |  |  | akЛqm  |  |  |  |
| а3  |  |  |  |  |  |  |
| а4  |  |  |  |  |  |  |
| …  |  |  |  |  |  |  |
| ам  |  |  |  |  |  |  |

Примеры

Описать машины Тьюринга, которые реализуют:

1. Счетчик четности. Выход машины Тьюринга равен 0 или 1 в зависимости от того, четно или нечетно число единиц в последовательности из 0 и 1, записанной на ленте машины Тьюринга. В конце последовательности стоит символ *B*. В начальном состоянии головка видит первый левый символ. 

2. Инверсию заданного слова в алфавите {0, 1} (0 заменяет на 1, а 1 – на 0). В начальном состоянии головка видит первый левый символ.



1. Прибавление единицы к заданному двоичному числу. В начальном состоянии головка видит первый правый символ.



4. вычесть единицу от заданного двоичного числа. В начальном состоянии головка видит первый правый символ.



1. «Переворачивание» заданного слова в алфавите {*a*, *b*, *c*}. В начальном состоянии головка видит первый правый символ.

Комментарий:

0- берем первую букву

4- проходим итоговое слово налево

5 проход пробелов между словами и запоминание символа для инверсии

6 7 8 проход пробелов вправо между словами с b a

9 10 11 проход символов итога вправо

12 13 14 проверка на последний символ

c b a

15 16 17 проход пробелов вправо между словами последний раз

18 19 20 проход символов итога вправо последний раз

1. Сложить два двоичных числа.