Практика 9-10 Машина Тьюринга

Формально – это расширение конечного автомата (КА).
В сущности все вычислительные процедуры могут быть разложены на элементарные операции. Это привело к построению математической модели, называемой машиной Тьюринга. Повторение элементарных операций, определенных этой машиной, достаточно для проведения любого возможного вычисления.

Особенности –

- У машины бесконечная память (хотя в каждый момент времени количество накопленной ею информации конечно, для этого нет никакой верхней границы.
- Машина никогда не ошибается, так как точно выполняет правила, установленные для ее работы.

Машина Тьюринга включает:

- *Внешний алфавит* $\mathbf{A=\{a_0, a_1,\ldots,a_n\}}$ –конечное множество символов. В символах этого алфавита информация вводится в машину. Машина перерабатывает эту информацию в новую.

- *Внутренний алфавит* $Q=\{q_0, q_1,\ldots,q_m, R, L, C, P, E\}$, где R-направо , L – налево, C – на месте, P- печать, E – стереть, символы $q_0, q_1,\ldots,q_m$ – конечное число состояний машины, причем q_1 – начальное состояние, q_0 –стоп-состояние.

- *Бесконечную в обе стороны ленту, представляющую память машины.* Эта память разбита на клетки. В каждую клетку может быть записана только одна буква. a_0 - пустая клетка, она всегда может появиться при движении вправо или влево, если закончится слово исходной информации $\mathbf{a_0, a_1,\ldots,a_n}$.

- Управляющую головку. Она передвигается вдоль ленты и может остановиться напротив какой – либо клетки и воспринимать записанный там символ исходного слова. В одном такте работы машины управляющая головка может сдвигаться только на одну клетку или оставаться на месте.

Окончание работы машины Тьюринга:

1. После конечного числа тактов машина останавливается в q_0 состоянии. При этом на ленте оказывается переработанная информация. Говорят, что машина применима к начальной информации I_1 и перерабатывает ее в результирующую информацию I_2.
2. Машина никогда не останавливается (не переходит в состояние q_0 – стоп-состояние). Говорят, что машина не применима к начальной информации I_1.

В каждом такте машина действует *по единой функциональной схеме:*

$$\boxed{a_i\, q_j \Rightarrow a_l \begin{Bmatrix} R \\ L \\ C \\ P \\ E \end{Bmatrix} q_s}$$

где $a_i, a_l \in A$,
$q_j, q_s \in Q$,
a_i – буква на ленте, обозреваемая управляющей головкой на данном такте
q_j – текущее состояние машины на данном такте (в общем случае не i-том и не j – том)
На каждом такте функциональной схемой вырабатывается команда, состоящая из трех элементов (правая часть формулы) :

1. Буква внешнего алфавита a_l, на которую заменяется обозреваемая буква a_i.
2. Адрес внешней памяти и дополнительные действия для выполнения на следующем такте R,L,C,P,E.(направо, налево, на месте, печать, стереть)
3. Следующее состояние машины.

Формирование правой части функциональной схемы происходит по командам, совокупность которых образует программу машины Тьюринга. Программу можно представить в виде двумерной таблицы, называемой тьюринговой функциональной схемой, в каждой клетке которой записываются отдельные команды. Работа машины Тьюринга полностью определяется ее программой.

Две машины Тьюринга с общей функциональной схемой идентичны, разные машины имеют разные программы, то есть разные функциональные схемы.

На дом.

1.Разобраться самостоятельно с примерами из доклада. Дополнительно создать функциональную схему к примеру1 , где столбцы – символы внешнего алфавита, строки – состояния.

2. Построить машину Тьюринга, реализующую алгоритм вычисления функции f(n) = n+2 в десятичной системе счисления. Построить таблицу ,реализующую функциональную схему.

Выводы

МТ – это абстрактная модель вычислений, то есть любой алгоритм в интуитивном смысле этого слова может быть представлен эквивалентной машиной в предложенной Тьюрингом модели вычислений. Вычислительная мощность МТ равна вычислительной мощности реальных компьютеров.

Отсюда также следует, что все компьютеры, независимо от мощности, архитектуры и других особенностей, эквивалентны с точки зрения принципиальной возможности решения алгоритмически разрешимых задач.

МТ уточняет понятие алгоритма и указывает путь решения так называемых массовых проблем. Однако МТ бывает неприменима к начальной информации (исходному слову алфавита). Такая же ситуация создается относительно некоторых задач, для решения которых не удается создать машины МТ. Например, распознавание выводимости в математической логике. (теорема Черча - проблема распознавания выводимости алгоритмически неразрешима.)

Это значит, что все проблемы, разрешимые с помощью МТ являются алгоритмически разрешимыми. А все проблемы, неразрешимые с помощью МТ, являются алгоритмически неразрешимыми.

Введение формальной модели алгоритмов было первым шагом, который привел к основанию теоретической информатики. А сам процесс был инициирован оригинальной работой Геделя, которая была первым доказательством существования математических проблем, которые не могут быть решены алгоритмически, то есть никаким методом! Именно результат, полученный Геделем, заставил ученых (Черч, Клини, Пост, Тьюринг) взяться за разработку формальных моделей интуитивного понятия алгоритма. Все эти модели оказались эквивалентными между собой. Результат этого опыта – тезис Черча-Тьюринга, который является единственной аксиомой теоретической информатики, а следовательно не может быть доказан. Он согласуется с нашим опытом. Но принципиальная возможность его опровержения существует. Пока этого не случилось.

Компьютер, несмотря на то, что он не обладает интеллектом, способен выполнить программу для некоторых конкретных входных данных, то есть решить частный случай. Поэтому мы имеем право говорить об автоматической (т.е. алгоритмической) разрешимости проблемы. Чтобы показать, что некоторая проблема автоматически разрешима, достаточно найти метод, решающий эту проблему – и записать этот метод в виде программы(или алгоритма). Именно это дает нам возможность не иметь формальное определение термина “алгоритм” (программа) для формулировки положительных утверждений об алгоритмической разрешимости проблем. Достаточно неформального описания алгоритма – и дальнейшей его реализации в виде компьютерной программы.

Потребность дать точное, формальное определение понятия “алгоритм”возникло, попытались доказать неразрешимость некоторых из проблем. Разные ученые стали вводить формальные определения этого понятия. И все эти определения оказались эквивалентны. И любой язык программирования можно рассматривать как выполнимую формализацию этой алгоритмической разрешимости. Но у языков программирования слишком сложные команды, поэтому они неудобны для доказательства несуществования алгоритмов решения конкретных проблем. Нам бы хотелось иметь такие модели алгоритмов, которые используют только элементарные команды, но при этом обладают полной вычислительной мощью программ, написанных на любом языке программирования высокого уровня.

Именно такой моделью является машина Тьюринга. Машина Тьюринга эквивалентна языкам программирования.

Машина Тьюринга – это абстрактная модель вычислений. При этом ее вычислительная мощность равна вычислительной мощности реальных компьютеров. Ячейки , из которых состоит лента, соответствуют регистру реального компьютера, символы рабочего алфавита соответствуют всем возможным компьютерным словам(возможному содержанию регистра). Лента рассматривается не только как вход, но и как память. Инструкции машины(команды) называются переходами.

Тезис Черча- Тьюринга говорит, что машина Тьюринга , которая всегда останавливается, является формализацией понятия алгоритм. Следовательно, все проблемы, разрешимые с помощью машин Тьюринга, являются алгоритмически(автоматически) разрешимыми. Наоборот, проблемы, неразрешимые с помощью машин Тьюринга, являются алгоритмически неразрешимыми. Тезис Черча-Тьюринга – единственная собственная аксиома информатики, и как всякая аксиома, никогда не может быть доказана.

Итак, была введена формальная модель алгоритмов. Этот процесс инициировала работа Геделя, в которой дано доказательство существования проблем, которые не могут быть решены алгоритмически – то есть никаким методом. Именно результат Геделя послужил мотивацией для разработки Черчем, Клини, Постом и Тьюрингом формальных моделей интуитивного понятия "алгоритм". Все эти модели оказались эквивалентными между собой.

Замечание: Не следует , однако, забывать, что машина Тьюринга – бесконечны, а компьютеры –конечны, поэтому любой компьютер может быть описан в пропозициональной логике, на которую не распространяется теорема Геделя о неполноте.

Тезис Черча-Тьюринга : Машина Тьюринга суть формализация понятия "алгоритм", то есть класс рекурсивных языков (разрешимых проблем принадлежности), соответствующих классу алгоритмически(автоматически) распознаваемых языков.

Машина Тьюринга не просто дала понятие алгоритма через класс рекурсивных функций, но связала это понятие с машинной арифметикой, сначала уточнив понятие алгоритма, а затем определило класс вычислимых функций.

Сложные процессы можно моделировать на простых устройствах. Алгоритм можно задать некоторой функциональной схемой и реализовать его в виде машины Тьюринга. Все известные на сегодняшний день алгоритмы могут быть заданы с помощью функциональных схем Тьюринга! (хотя это не доказать).

Вернитесь к первым лекциям, где мы говорили об алгоритмически неразрешимых задачах. Уже там мы записали, что машина Тьюринга способна вычислить любую вычислимую функцию. Но существуют некоторые функции, которые не могут быть вычислены машиной Тьюринга.
Например, ни одна машина не способна определить, возвратит ли данная конкретная программа ответ на конкретные входные данные или будет работать до бесконечности.

Решить самостоятельно:

Пусть начальная конфигурация имеет вид $a_0\ a_1\ a_2\ a_2\ a_0$. Машина Тьюринга
q_1
управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_2 C q_1
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?

45

Пусть начальная конфигурация имеет вид a_0 a_1 a_0. Машина Тьюринга
q_1
управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_2 C q_1
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?

46

Пусть начальная конфигурация имеет вид a_0 a_1 a_2 a_2 a_0. Машина Тьюринга
q1
управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_2 C q_1
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?

47

Пусть начальная конфигурация имеет вид $a_0\ \underset{q_1}{a_1}\ a_2$. Машина Тьюринга управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_2 C q_1
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?

48

Пусть начальная конфигурация имеет вид $\underset{q_2}{a_0}\ a_1\ a_2\ a_0$. Машина Тьюринга управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_2 C q_1
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?

49

Пусть начальная конфигурация имеет вид a_0 a_2 a_1 a_1. Машина Тьюринга
q_2
управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_2 C q_1
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?

149

Пусть начальная конфигурация имеет вид a_0 a_1 a_2 a_2. Машина Тьюринга
q_3
управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_1 R q_0
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?

148

Пусть начальная конфигурация имеет вид a_0 a_2 a_1 a_2. Машина Тьюринга
q_3
управляется функциональной схемой, представленной ниже:

Текущее состояние машины на очередном такте	a_0	a_1	a_2
q_1	a_2 L q_3	a_1 R q_2	a_2 L q_1
q_2	a_1 C q_0	a_2 C q_1	a_1 C q_2
q_3	a_0 R q_0	a_1 R q_4	a_2 L q_0
q_4	a_1 C q_3	a_0 R q_4	a_2 R q_4

q_1 – начальное состояние;
q_0 – конечное состояние
Каков конечный результат? Применима ли машина к исходной информации?
149
Задан алфавит входного слова $\Sigma = \{0,1,2,3,4,5,6,7,8,9\}$
Входное слово - неотрицательное число в десятичной СС (последовательность из десятичных цифр). Требуется получить на ленте запись числа, которое на 1 больше входного числа.
150
$\Sigma=\{a,b,c\}$ 1 Перенести символ непустого слова W в его конец.
2.Сжать слово W, удалив из него первое вхождение символа a , если такой естью
151
$\Sigma=\{a,b\}$. 1)Удалить из слова W второй символ, если такой есть.
2) Если W – непустое слово, то за его первым символом вставить символ r.