Формализация понятия «Алгоритм»

Информатика

Постановка проблемы

Определение алгоритма, можно назвать понятием алгоритма в интуитивном смысле.

Свойства алгоритмов следует называть **эмпирическими**. Однако, как фундаментальное научное понятие алгоритм требует более обстоятельного изучения. Оно невозможно без уточнения понятия «алгоритм», более строгого его описания или, как еще говорят, без его **формализации.**

Известно несколько подходов к формализации понятия «алгоритм»:

- теория конечных и бесконечных автоматов;
- теория вычислимых (рекурсивных) функций;
- λ-исчисление Черча.

Постановка проблемы

Главная цель формализации понятия алгоритма такова: подойти к решению проблемы алгоритмической разрешимости различных математических задач, т.е. ответить на вопрос, может ли быть построен алгоритм, приводящий к решению задачи.

Вместе с тем, формально, определенный любым из известных способов, алгоритм не может в практическом программировании заменить то, что мы называли алгоритмами в предыдущей лекции. Основная причина состоит в том, что формальное определение резко сужает круг рассматриваемых задач, делая многие практически важные задачи недоступными для рассмотрения.

История

Машины Поста и Тьюринга, предназначенные для доказательств различных утверждений о свойствах программ для них, были предложены независимо друг от друга в 1936 г. американским математиком Эмилем Постом и английским математиком Алланом Тьюрингом.

Эти машины представляют собой универсальных исполнителей, являющихся полностью детерминированными, позволяющих «вводить» начальные данные, и после выполнения программ «читать» результат. Машина Поста менее популярна, хотя она значительно проще машины Тьюринга. С ее помощью можно вести обучение первым навыкам составления программ для ЭВМ.

Машина Поста

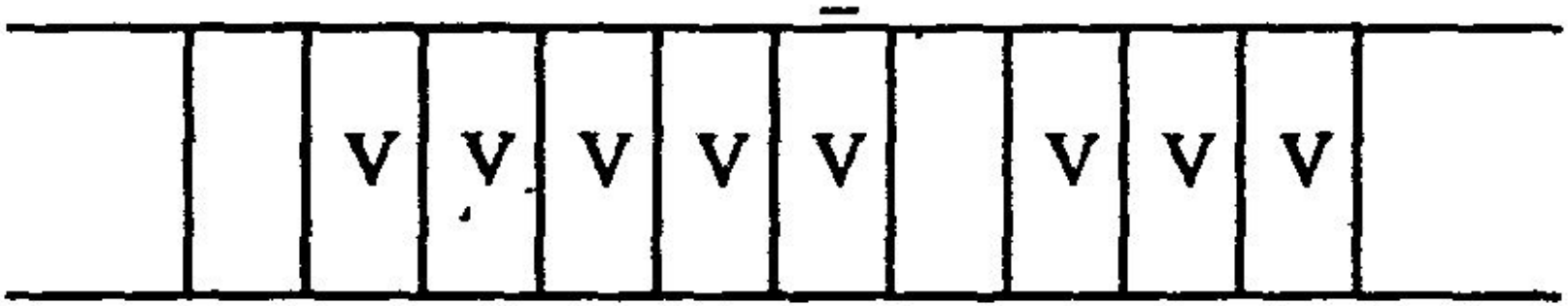
Абстрактная машина Поста представляет собой бесконечную ленту, разделенную на одинаковые клетки, каждая из которых может быть либо пустой, либо заполненной меткой «V», и головки, которая может перемещаться вдоль ленты на одну клетку вправо или влево, наносить в клетку ленты метку, если этой метки там ранее не было, стирать метку, если она была, или проверять наличие в клетке метки.

Информация о заполненных метками клетках ленты характеризует состояние ленты, которое может меняться в процессе работы машины.

Машина Поста (2)

В каждый момент времени головка («-») находится над одной из клеток ленты и, как говорят, обозревает ее.

Информация о местоположения головки вместе с состоянием ленты характеризует состояние машины Поста.



Команда машины Поста

Структура команды:

n Km,

где

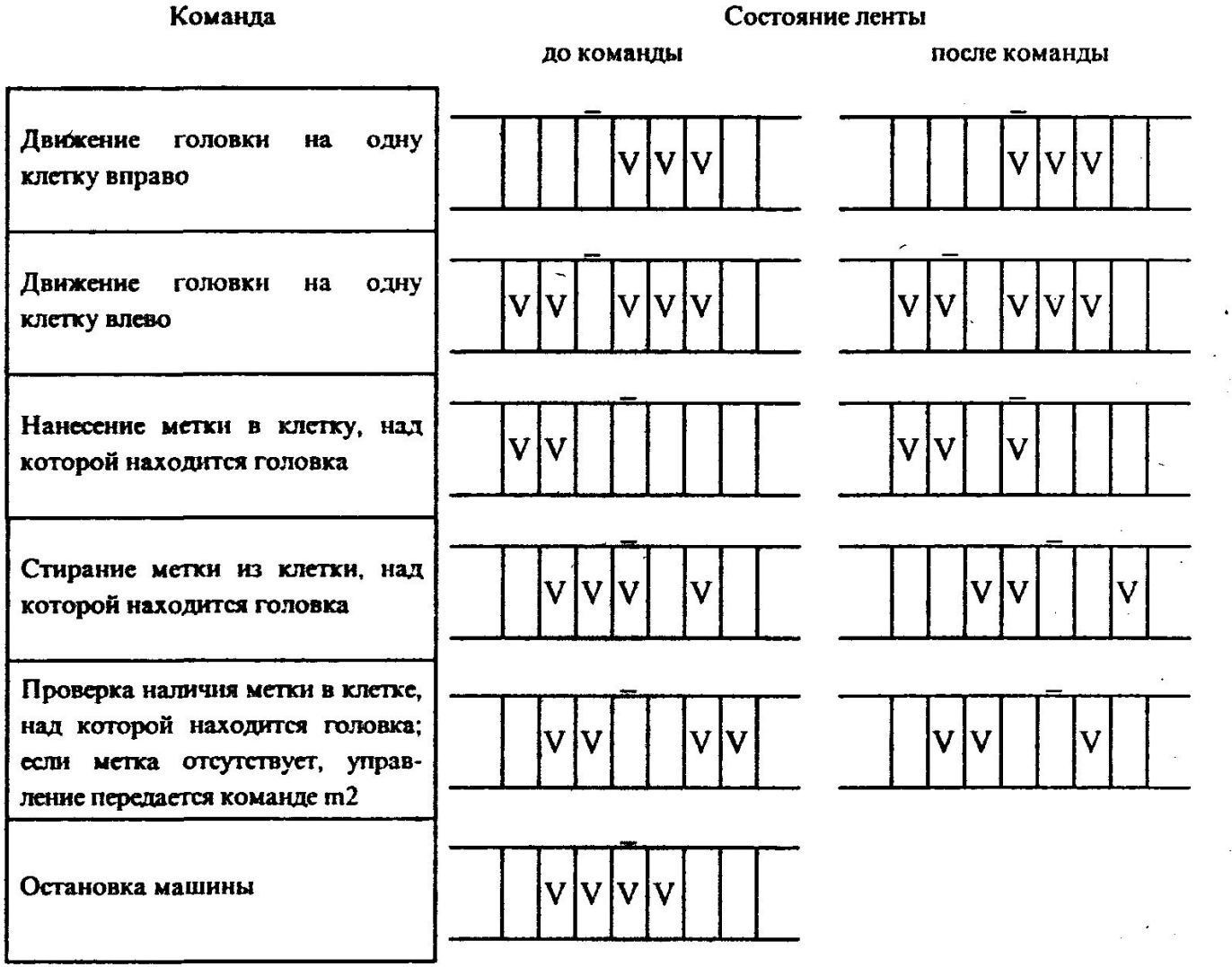
n - порядковый номер команды,

K-действие, выполняемое головкой,

m - номер следующей команды, подлежащей выполнению.

Существует всего шесть команд машины Поста.

Команды машины Поста



Машина Поста

Программой для машины Поста будем называть непустой список команд, такой что 1) на *n*-м месте команда с номером *n*; 2) номер *m* каждой команды совпадает с номером какой-либо команды списка.

С точки зрения свойств алгоритмов, изучаемых с помощью машины Поста, наибольший интерес представляют причины останова машины при выполнении программы:

1) останов по команде «стоп»; такой останов называется **результативным** и указывает на корректность алгоритма (программы);

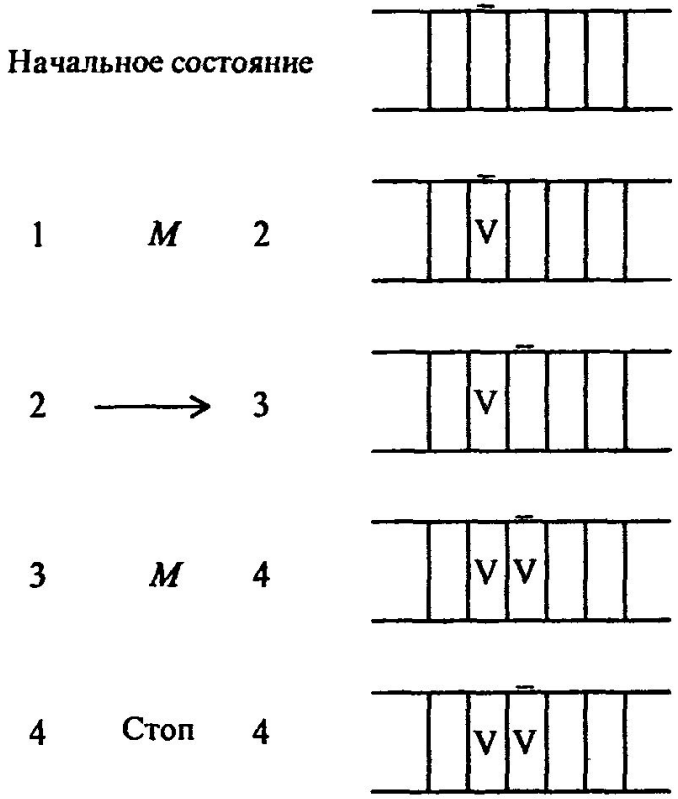
2) останов при выполнении недопустимой команды; в этом случае останов называется **безрезультативным**;

3) машина не останавливается никогда; в этом и в предыдущем случае мы имеем дело с **некорректным алгоритмом** (программой).

Будем понимать под начальным состояние расположение головки против пустой клетки левее самой левой метки на ленте.

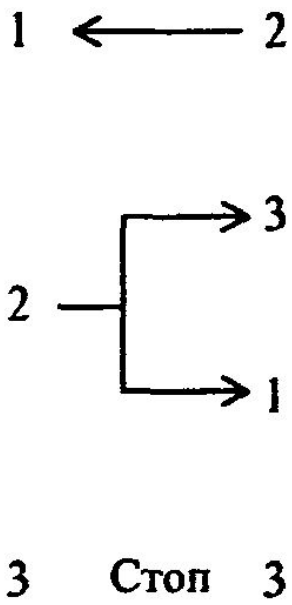
Машина Поста. Пример №1.

- Пусть задано исходное состояние головки и требуется на пустой ленте написать две метки: одну в секцию под головкой, вторую справа от нее.



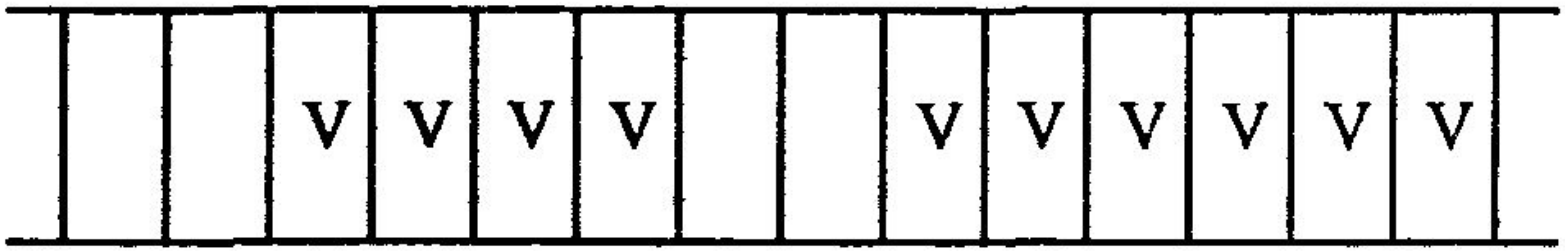
Машина Поста. Пример №2.

- Покажем, как можно воспользоваться командой условного перехода для организации циклического процесса. Пусть на ленте имеется запись из нескольких меток подряд и головка находится над самой крайней меткой справа. Требуется перевести головку влево до первой пустой позиции. .



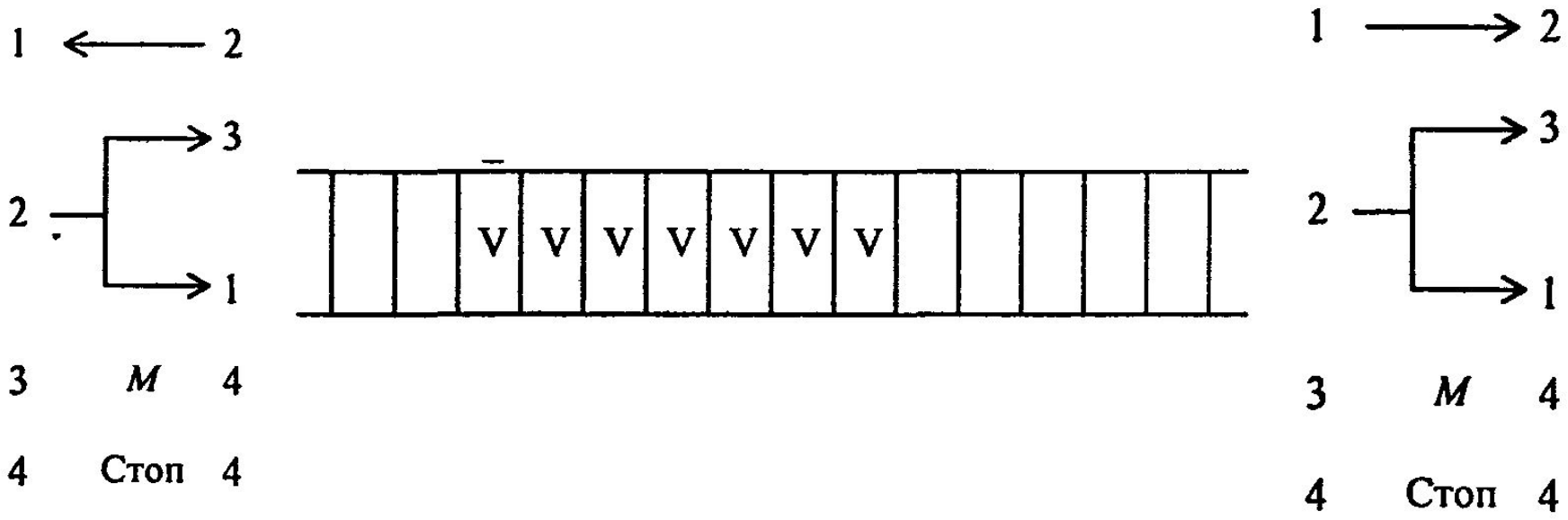
Машина Поста. Пример №3.

- Представлении чисел на ленте машины Поста и выполнении операций над ними.
- Число *k* представляется на ленте машины Поста идущими подряд *k* + 1 метками (одна метка означает число «О»). Между двумя числами делается интервал как минимум из одной пустой секции на ленте. Например, запись чисел 3 и 5 на ленте машины Поста будет выглядеть так:
- Обратим внимание, что используемая в машине Поста система записи чисел является непозиционной.



Машина Поста. Пример №3.

- Программа для прибавления к произвольному числу единицы. Предположим, что на ленте записано только одно число и головка находится над одной из клеток, в которой находится метка, принадлежащая этому числу.



Машина Поста.

Машину Поста можно рассматривать как упрощенную модель ЭВМ.

В самом деле, как ЭВМ, так и машина Поста имеют:

- неделимые носители информации (клетки - биты), которые могут быть заполненными или незаполненными;
- ограниченный набор элементарных действий - команд, каждая из которых выполняется за один такт (шаг).

Обе машины работают на основе программы. Однако, в машине Поста информация располагается линейно и читается подряд, а в ЭВМ можно читать информацию по адресу; набор команд ЭВМ значительно шире и выразительнее, чем команды машины Поста и т.д.

Машина Тьюринга

Машина Тьюринга

- Машина Тьюринга (МТ) состоит из счетной ленты, читающей и пишущей головки, лентопротяжного механизма и операционного исполнительного устройства, которое может находиться в одном из дискретных состояний q_0, q_1, ..., q_s, принадлежащих некоторой конечной совокупности. При этом q_0 называется **начальным состоянием**.
- Читающая и пишущая головка может читать буквы рабочего алфавита $A = \{a_0, a_1, ..., a_t\}$, стирать их и печатать. Каждая ячейка ленты в каждый момент времени занята буквой из множества A. Чаще всего встречается буква a_0 - «пробел». Головка находится в каждый момент времени над некоторой ячейкой ленты - **текущей рабочей ячейкой**. Лентопротяжный механизм может перемещать ленту так, что головка оказывается над соседней ячейкой ленты. При этом возможна ситуация выхода за левый край ленты (ЛК), которая является аварийной, или машинного останова (МО), когда машина выполняет предписание об остановке.

Машина Тьюринга

- Порядок работы МТ (с рабочим алфавитом a_0, a_1, ..., a_t и состояниями q_0, q_1, ..., q_s) описывается таблицей машины Тьюринга. Эта таблица является матрицей с четырьмя столбцами и $(s + 1)$ $(t + 1)$ строками.
- Каждая строка имеет вид

$$q_i\, a_j\, v_{ij}\, q_{ij}, \quad 0 \le i \le s, \quad 0 \le j \le t, \quad q_{ij} \in \{q_0, q_1, \ldots, q_s\}.$$

Машина Тьюринга

$$q_i\, a_j\, v_{ij}\, q_{ij}, \quad 0 \le i \le s, \quad 0 \le j \le t, \quad q_{ij} \in \{q_0, q_1, \ldots, q_s\}.$$

Здесь:

- v_{ij} обозначен элемент объединения алфавита $\{a_0, a_1, \ldots, a_t\}$ и множества предписаний для лентопротяжного механизма: l - переместить ленту влево, r -переместить ленту вправо, s - остановить машину;
- v_{ij} - действие МТ, состоящее либо в занесении в ячейку ленты символа алфавита $a_0, a_1, \ldots, a_t$, либо в движении головки, либо в останове машины; q_{ij} является последующим состоянием.

Работа машины Тьюринга

МТ работает согласно следующим правилам:

- если МТ находится в состоянии q_i, головка прочитывает символ 0 в рабочей ячейке. Пусть строка $q_i \ a_j \ v_{ij} \ q_{ij}$, начинающаяся с символов q_i, a_j, встречается только один раз в таблице.
- Если v_{ij} - буква рабочего алфавита, то головка стирает содержимое рабочей ячейки и заносит туда эту букву.
- Если v_{ij} - команда *r* или *l* для лентопротяжного механизма, то лента сдвигается на одну ячейку вправо или влево (если не происходит выход за левый край ленты) соответственно.
- Если $v_{ij} = s$, то происходит машинный останов.

Машина Тьюринга. Пример №1.

Алгоритм прибавления единицы к числу n в десятичной системе счисления. Дана десятичная запись числа n; требуется получить десятичную запись числа $n + 1$.

Очевидно, что внешний алфавит МТ должен состоять из десяти цифр 0,1,2,3,4,5,6,7,8,9 и символа пробела _. Эти цифры записывают по одной в ячейке (подряд, без пропусков).

Оказывается достаточным иметь два внутренних состояния машины: q_1 и q_2.

Предположим, что в начальный момент головка находится над одной из цифр числа, а машина находится в состоянии q_1. Тогда задача может быть решена в два этапа: движения головки к цифре единиц числа (во внутреннем состоянии q_1) и замене этой цифры на единицу большую (с учетом переноса 1 в следующий разряд, если это 9, во внутреннем состоянии q_2. Соответствующая схема МТ может иметь вид

Машина Тьюринга. Пример №1.

a_i	q_i	
	q_1	q_2
0	0Пq_1	1Сq_2
1	1Пq_1	2Сq_2
2	2Пq_1	3Сq_2
3	3Пq_1	4Сq_2
4	4Пq_1	5Сq_2
5	5Пq_1	6Сq_2
6	6Пq_1	7Сq_2
7	7Пq_1	8Сq_2
8	8Пq_1	9Сq_2
9	9Пq_1	0Сq_2
-	-Лq_1	1Сq_2

Нормальные алгоритмы Маркова

Рассмотрим некоторые понятия ассоциативного исчисления. Пусть имеется **алфавит** (конечный набор различных символов). Составляющие его символы будем называть **буквами**. Любая конечная последовательность букв алфавита (линейный их ряд) называется **словом** в этом алфавите.

Рассмотрим два слова *N* и *M* в некотором алфавите *A*. Если *N* является частью *M,* то говорят, что *N* входит в *M*.

Зададим в некотором алфавите конечную систему подстановок *N - M, S - T,...,* где *N, M, S, T,...* - слова в этом алфавите. Любую подстановку *N-M* можно применить к некоторому слову *K* следующим способом: если в *K* имеется одно или несколько вхождений слова *N,* то любое из них может быть заменено словом *M,* и, наоборот, если имеется вхождение *M,* то его можно заменить словом *N*.

Например, в алфавите $A = \{a, b, c\}$ имеются слова $N = ab$, $M = bcb$, $K = abcbcbab$, Заменив в слове *K* слово *N* на *M,* получим *bcbcbcbab* или *abcbcbbcb,* и, наоборот, заменив *M* на *N,* получим *aabcbab* или *abcabab*.

Нормальные алгоритмы Маркова

Подстановка *ab - bcb* недопустима к слову *bacb,* так как ни *ab,* ни *bcb* не входит в это слово. К полученным с помощью допустимых подстановок словам можно снова применить допустимые подстановки и т.д. Совокупность всех слов в данном алфавите вместе с системой допустимых подстановок называют **ассоциативным исчислением.** Чтобы задать ассоциативное исчисление, достаточно задать алфавит и систему подстановок.

Слова *P1* и *P2* в некотором ассоциативном исчислении называются **смежными**, если одно из них может быть преобразовано в другое однократным применением допустимой подстановки.

Последовательность слов *P, P1, P2, ..., M* называется **дедуктивной цепочкой**, ведущей от слова *P к* слову *M,* если каждое из двух рядом стоящих слов этой цепочки - смежное.

Слова *P* и *M* называют **эквивалентными**, если существует цепочка от *P* к *M* и обратно.

Нормальные алгоритмы Маркова. Пример.

Алфавит	Подстановки	
{a, b, c, d, e}	*ac - ca,*	*abac - abace*
	ad - da;	*eca - ae*
	bc - cb;	*eda - be*
	bd - db;	*edb - be*

Нормальные алгоритмы Маркова

Нормальные алгоритмы Маркова являются не только средством теоретических построений, но и основой специализированного языка программирования, применяемого как язык символьных преобразований при разработке систем искусственного интеллекта. Это один из немногих языков, разработанных в России и получивших известность во всем мире.

Существует строгое доказательство того, что по возможностям преобразования нормальные алгоритмы Маркова эквивалентны машинам Тьюринга.

Рекурсивные функции

Еще одним подходом к проблеме формализации понятия алгоритма являются, так называемые, **рекурсивные функции**.

Рекурсией называется способ задания функции, при котором значение функции при определенном значении аргументов выражается через уже заданные значения функции при других значениях аргументов. Применение рекурсивных функций в теории алгоритмов основано на идее нумерации слов в произвольном алфавите последовательными натуральными числами. Таким образом любой алгоритм можно свести к вычислению значений некоторой целочисленной функции при целочисленных значениях аргументов.

Рекурсивные функции. Основные понятия.

Пусть X, Y - два множества. Частичной функцией (или отображением) из X в Y будем называть пару $<D(f), f>$, состоящую из подмножества $D(f) \subset X$ (называемого областью определения f) и отображения f: $D(f) \rightarrow Y$. Если $D(f)$ пусто, то f нигде не определена. Будем считать, что существует единственная нигде не определенная частичная функция.

Рекурсивные функции. Основные понятия.

Через N будем обозначать множество натуральных чисел. Через $(N)^n$ (при $n \geq 1$) будем обозначать n-кратное декартово произведение N на себя, т.е. множество упорядоченных n-ок $(x_1 \ldots, x_n)$, $x_i \subset N$. Основным объектом дальнейших построений будут частичные функции из $(N)^m$ в $(N)^n$ для различных m и n.