Соловьев С.Ю.

soloviev@glossary.ru

Алгоритмы и Алгоритмические языки

www.park.glossary.ru/pascal/

Лекция No.2

2022

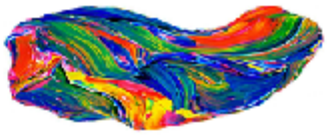
Напоминание

Задача — свойства — единственность

Алгоритм:
- способы задания: словесное описание; блок-схема
- свойства: корректность; конечность; сложность

слово → слово

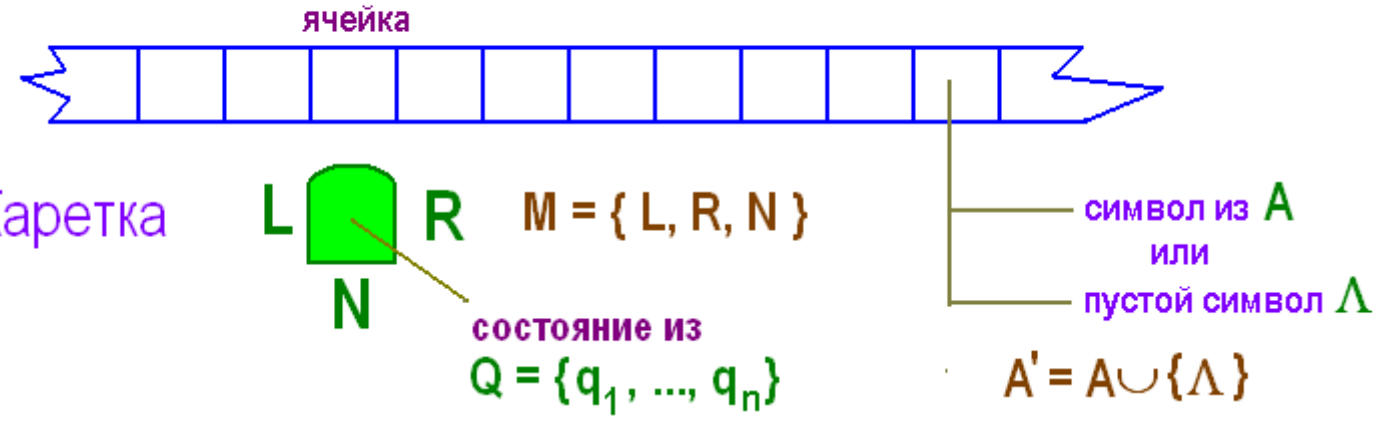
Машина Тьюринга



Машина Тьюринга (1936)

< ЛЕНТА, КАРЕТКА, ПРОГРАММА>

- Бесконечная лента состоит из ячеек



Каретка выполняет правила вида **a, q → a', q', m**

Если в ячейке находится символ **a**,
и каретка находится в состоянии **q**,
то 1. записать в ячейку символ **a'**,
2. изменить состояние на **q'**,
3. выполнить движение **m**.

- Программа – мн-во **P** правил для каретки, обладающее свойством детерминированности.

Свойство детерминированности

Если $a, q \to a', q', m'$ – правило из P,

и $a, q \to a'', q'', m''$ – правило из P,

то $a' = a'', \ q' = q'', \ m' = m''$

Соглашения

(1) Способ кодирования входного слова

(2) q_1 – начальное состояние каретки

(3) Начальное положение каретки на ленте

(4) Способ кодирования выходного слова

Соглашения

(5) Выполнение правила **a, q → a, q, N**

означает СТОП.

(6) Программу P можно задавать в виде таблицы:

	A			
	Λ	...	a	...
q_1				
...				
q			a',q',m	
...				
q_n				

a, q → a', q', m

(7) см. далее

(8) см. далее

(9) см. далее

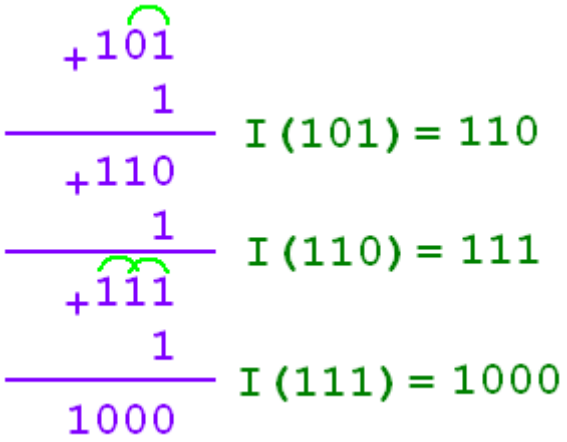
Пример МТ / 1

МТ, которая увеличивает заданное двоичное число на 1.

слово из $\{0,1\}^+$,
которое начинается символом 1:

$\delta_1\delta_2 \ldots \delta_n$, $\delta_1 = 1$, $\delta_i \in \{0,1\}$, $i > 1$.

N_{10}	N_2
1	1
2	10
3	11
4	100
5	101
...	
255	11111111
256	100000000



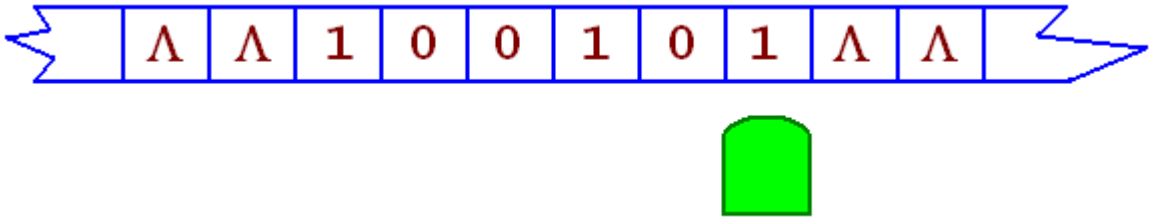
Пример МТ / 2

Св-во 1. $I(\delta_1 \ldots \delta_{n-1}0) = \delta_1\delta_2 \ldots \delta_{n-1}1, \quad n > 0$

Св-во 2. $I(\delta_1 \ldots \delta_{n-1}1) = I(\delta_1\delta_2 \ldots \delta_{n-1})0, \quad n > 0$

Св-во 3. $I(\,) = 1, \quad n = 0$

(3) Каретка обозревает последний знак.



P	Λ	0	1
q_1	$1\,q_2\,N$ C3	$1\,q_2\,N$ C1	$0\,q_1\,L$ C2
q_2	$\Lambda\,q_2\,N$	$0\,q_2\,N$	$1\,q_2\,N$

} Соглашение (5)

Пример МТ / 3

P	Λ	0	1
q_1	$1\,q_2\,N$ C3	$1\,q_2\,N$ C1	$0\,q_1\,L$ C2
q_2	$\Lambda\,q_2\,N$	$0\,q_2\,N$	$1\,q_2\,N$

Последовательность действий

q_1 : Λ100101Λ

$1\,q_1 \rightarrow 0\,q_1\,L$

q_1 : Λ100100Λ

$0\,q_1 \rightarrow 1\,q_2\,N$

q_2 : Λ100110Λ

$1\,q_2 \rightarrow 1\,q_2\,N$ стоп

Пример МТ / модификация

Увеличить на 1 и вернуть каретку на правый край.

P	Λ	0	1
q_1	$1\,q_2\,N$ C3	$1\,q_2\,N$ C1	$0\,q_1\,L$ C2
q_2	$\Lambda\,q_3\,L$ коррект.	$0\,q_2\,R$ двигаться вправо	$1\,q_2\,R$
q_3	$\Lambda\,q_3\,N$	$0\,q_3\,N$	$1\,q_3\,N$

$\Lambda 100110001\blacksquare 0000\Lambda$

} Соглашение (5)

Соглашения +

(7) ! – состояние останова

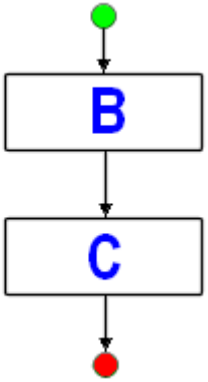
(8) N – не пишем

(9) Если правило не изменяет символ, то этот символ не пишем.

P	Λ	0	1
q_1	$1\,q_2$ C3	$1\,q_2$ C1	$0\,q_1\,L$ C2
q_2	! L коррект.	$q_2\,R$ двигаться вправо	$q_2\,R$

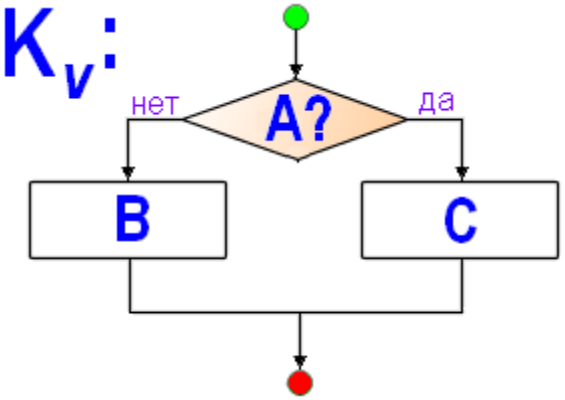
Композиции алгоритмов

K_s:



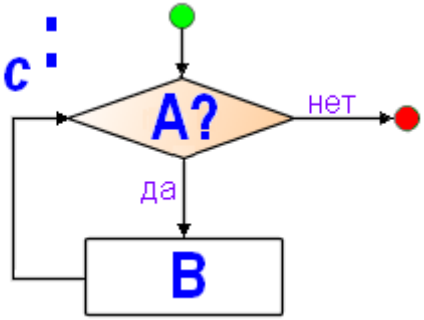
Если $\exists MT_B$ и $\exists MT_C$, то $\exists MT_{Ks}$

K_v:



Если $\exists MT_A$, $\exists MT_B$ и $\exists MT_C$, то $\exists MT_{Kv}$

K_c:



Если $\exists MT_A$ и $\exists MT_B$, то $\exists MT_{Kc}$

Тезис Тьюринга

Всякий алгоритм может быть реализован машиной Тьюринга.

Программа в виде слова

P	Λ	0	1
q_1	$1\ q_2$ С3	$1\ q_2$ С1	$0\ q_1 L$ С2
q_2	! L (коррект.)	$q_2 R$ (двигаться вправо)	$q_2 R$

$\Lambda\ q_1 \to 1\ q_2$
$0\ q_1 \to 1\ q_2$
$1\ q_1 \to 0\ q_1\ L$
$\Lambda\ q_2 \to !\ L$
$0\ q_2 \to q_2\ R$
$1\ q_1 \to q_2\ R$

Алфавит = { λ, 0, 1, q_1, q_2, !, L,R, →, ▪ } (λ вместо Λ)

Запись P =

$\lambda q_1 \to 1\ q_2 \blacksquare 0\ q_1 \to 1\ q_2 \blacksquare 1\ q_1 \to 0\ q_1\ L \blacksquare\ \lambda q_2 \to !\ L \blacksquare 0\ q_2 \to q_2\ R \blacksquare 1\ q_1 \to q_2\ R \blacksquare$

длина = 43

Любую программу МТ можно представить в виде слова.

Существует универсальная машина Тьюринга

Пусть P -- программа произвольной МТ: вх.слово → вых.слово

Нормальные алгоритмы Маркова (1954)

Алгоритм: Вх.слово $\rightarrow$ Вых.слово

Нормальный алгоритм Маркова (в алфавите A) – конечная упорядоченная последовательность правил

$$П_1, \ldots, П_k, \ldots, П_n,$$

в которой каждое правило $П_k$ имеет вид:

$$L_k \rightarrow R_k \quad \text{или} \quad L_k \mapsto R_k$$

для некоторых L_k и R_k из A^*.

Если L_k – пустое слово, то $П_k$ есть $\rightarrow R_k$ или $\mapsto R_k$.

Если R_k – пустое слово, то $П_k$ есть $L_k \rightarrow$ или $L_k \mapsto$.

Нормальные алгоритмы Маркова – 2

Пусть L и Y – слова

«**L** входит в **Y**», если $Y = Y_1 L Y_2$ (можно представить)

L = пра, **Y =** прапрапрадед,

Y = Lпрапрадед **=** пра**L**прадед **=** прапра**L**дед

Вхождение = самое левое вхождение,
для которого длина Y_1 минимальна.

Нормальные алгоритмы Маркова – 3

Пусть **П** – правило **L → R** или **L ↦ R.**

Правило П неприменимо к слову **Y,** если $Y \neq Y_1 L Y_2$.

Правило П применимо к слову **Y,** если L входит в **Y,**

т.е. $Y = Y_1 L Y_2$, и **результат применения** есть $Y_1 R Y_2$.

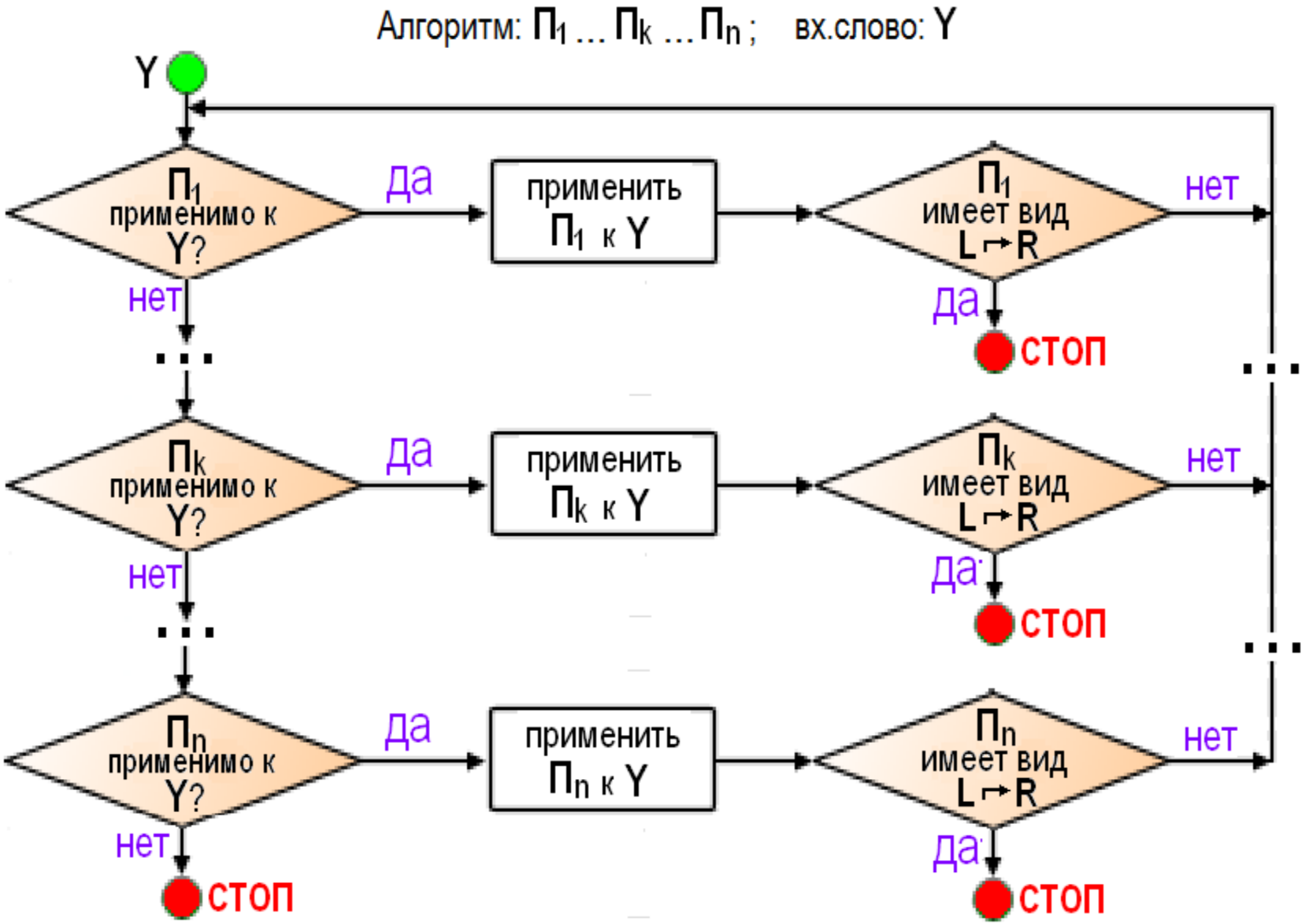
Если **П:** нос → горло и **Y** = обороноспособность, то

П применимо к **Y,** рез-т: $Y_п$ = оборогорлопособность.

П применимо к $Y_п$, рез-т: $Y_{пп}$ = оборогорлопособгорлоть.

П неприменимо к $Y_{пп}$.

Выполнение нормального алгоритма



Примеры

Кодировка.
Заменить a,b,c на 1, 10, 100.

(1) **a → 1**
(2) **b → 10**
(3) **c → 100**

accb ⇒(1)⇒
1ccb ⇒(2)⇒
1cc10 ⇒(3)⇒
1100c10 ⇒(3)⇒
110010010 стоп

Декодировка (правильная).

(1) **100 → c**
(2) **10 → b**
(3) **1 → a**

101001 ⇒(1)⇒
10c1 ⇒(2)⇒
bc1 ⇒(3)⇒
bca стоп

Декодировка (неправильная).

(1) **10 → b**
(2) **100 → c**
(3) **1 → a**

101001 ⇒(1)⇒
b1001 ⇒(1)⇒
bb01 ⇒(3)⇒
bb0a стоп

Применимость

Нормальный алгоритм Маркова

- **применим** к входному слову, если он останавливается;
- **неприменим** к входному слову, если он не останавливается.

Достаточные условия применимости

1. Если для всех правил вида $L \to R$ имеет место: $L \in (A_1)^*$, $R \in (A_2)^*$ и $A_1 \cap A_2 = \varnothing$.
2. Если для всех правил вида $L \to R$ имеет место: длина(**L**) > длина(**R**).

Алгоритмы самоприменимые и несамоприменимые

Применимость алгоритма к собственной записи.

P(запись **P**)

останавливается

(1) **#/// → /#**
(2) **# ↦ #**
(3) **→ #**

самоприменимый

#/// → /#,# ↦ #,→ # ⇒(1)⇒
/# → /#,# ↦ #,→ # ⇒(2)⇒
/# → /#,# ↦ #,→ # стоп

(1) **→ a**
(2) **b ↦ b**

несамоприменимый

→ a,b ↦ b ⇒(1)⇒
a→ a,b ↦ b ⇒(1)⇒
aa→ a,b ↦ b и т.д.

не останавливается

Алгоритмическая неразрешимость

Утв. Не существует алгоритм **S**, который на основании записи любого алгоритма **P** вычисляет значение **C**, если **P** – самоприменим, и значение **H**, если **P** – несамоприменим.

Если ***из Y следует Z,***
и ***Z – ложно,***
то ***Y – ложно.***

Y = алгоритм S существует
Z = алгоритм K существует

Доказательство—1

Часть 1. Построим алгоритм **B**, который,

-- получив на входе символ **C**, зацикливается;

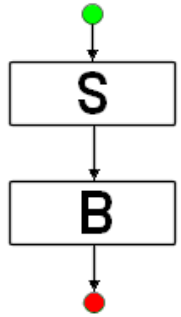
-- получив на входе символ **H**, останавливается.

B: **CH → C**
C → CH
H ↦ H

Алгоритм **B** существует.

Часть 2. Предположим алгоритм **S** существует.

Часть 3. Определим алгоритм **K** след.образом:



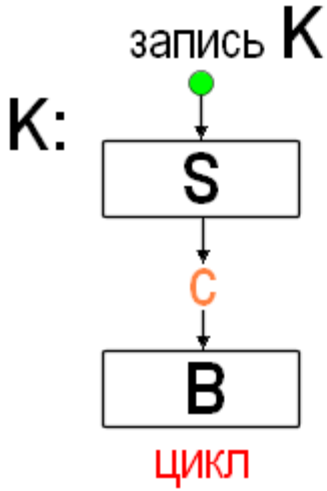
Алгоритм **K** существует как композиция существующих алгоритмов.

(Т.е. из существования **S** следует существование **K**.)

Доказательство—2

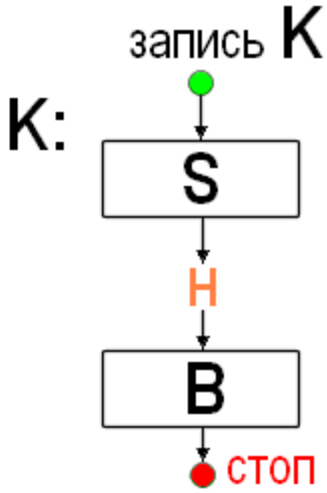
Часть 4. Исследуем алгоритм **К** на самоприменимость.

Пусть **К** самоприменим.
Тогда:



т.е. **К** не является самоприменимым

Пусть **К** несамоприменим.
Тогда:



т.е. **К** не является несамоприменимым

К не существует. Следовательно, **S** не существует.

Компьютеры / ЭВМ

исполнитель алгоритмов

ЭНИАК 1946 (США)
МЭСМ 1950 (СССР)

< **ПАМЯТЬ, ПРОЦЕССОР, ВНЕШНИЕ УСТРОЙСТВА** >

Обзор лекции No.2

Соглашения об использовании МТ

Композиция алгоритмов

Тезис Тьюринга

Универсальная МТ

Нормальные алгоритмы Маркова

Применимость правила к слову

Выполнение НАМ

Достаточные условия применимости НАМ

Самоприменимые и несамоприменимые алгоритмы

Алгоритмически неразрешимые задачи

--- Конец лекции No. 2 ---