ПРЕДСТАВИМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ. МОДЕЛЬ Э.Л. ПОСТА

ПЛАН ЛЕКЦИИ

1. Вклад Поста в теорию алгоритмов
2. Постановка задачи по Посту
3. Система команд и программа машины Поста
4. Финитный 1-процесс
5. Машина Поста
6. Инкремент числа на машине Поста.
7. Формулировка гипотезы Поста
8. Вычислимые функции и тезис Поста
9. Постулат Поста

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
1. Вклад Э.Л. Поста в теорию алгоритмов

Статья Эмиля Л. Поста “*Финитные комбинаторные процессы, формулировка 1*” была опубликована в издании “*Журнал символической логики*”, т. 1, № 3, 1936 г.

Emil L. Post: “*Finite combinatory processes – Formulation 1*”.

В предисловии к статье редактором журнала было отмечено, что статью Е.Л. Поста *необходимо сопоставить со статьёй* А.М. Тьюринга, которая была опубликована ранее, в мае 1936 г. Подчёркивалось, что *статьи написаны независимо друг от друга*.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
2. Постановка задачи по Посту

Допущения:

- имеем общую проблему, состоящую из множества конкретных проблем;
- решением общей проблемы будет такое решение, которое представляет ответ для каждой конкретной проблемы;
- проблемы сформулированы в пространстве символов, в котором должна происходить работа (Автором представлено в виде последовательности ящиков, с содержимым которых манипулирует клерк);
- имеется конечный набор инструментов (инструкций клерку), зафиксированный и не изменяемый для всех проблем.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
3. Система команд и программа машины Поста

Постом были предложены следующие операции (система команд):

- (a) – пометить ячейку (ящик), если он не помечен (пуст);
- (b) – удалить отметку, если текущий ящик помечен;
- (c) – сместиться на одну позицию (ящик) вправо;
- (d) – сместиться на одну позицию (ящик) влево;
- (e) – определить, помечена ли текущая ячейка.

Программа, по Посту, представляется конечным числом инструкций вида

- (A) – совершить операцию O_i, $O_i = \{a, b, c, d\}$
- (B) – совершить операцию e и, в зависимости от результата проверки, “да” или “нет”, перейти к выполнению операции $J^{\nabla}{}_i$ или $J^{\nabla\nabla}{}_i$.
- (C) – остановка.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
4. Финитный 1-процесс (1)

- Набор инструкций применим к данной *общей* проблеме, если его применение к частной проблеме *не требует операции* (*a*), когда ячейка помечена или *операции* (*b*), если она не помечена.
- Ход выполнения инструкций трактуется как детерминированный процесс в ходе применения к каждой конкретной проблеме, который окончится, когда дойдёт до (*c*).

  Набор инструкций задаёт ***финитный 1-процесс*** в связи с данной проблемой, если он применим к этой проблеме и ***если определяемый набором процесс заканчивается для каждой конкретной проблемы***: финитный 1-процесс, связанный с решением некоторой проблемы, будем называть ***1-решением*** этой проблемы, если даваемый им для каждой конкретной проблемы ответ является правильным.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
4. Финитный 1-процесс (2)

Пусть между классом целых чисел и классом конкретных проблем установлено взаимно-однозначное соответствие.

Общую проблему будем называть *1-заданной*, если *есть финитный 1-процесс*, который в результате применения к положительным числам, *выдаёт номера проблем*, образующих общую проблему.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
5. Машина Поста (1)

Обозначим команды:

- (*a*): **X**;
- (*b*): **§**;
- (*c*): $\Rightarrow$;
- (*d*): $\Leftarrow$;
- (*e*): **?** $\to$ **ПУСТО** $\to$ **ПОМЕЧЕНО**.

Состояние ленты – функция, которая каждому числу (номеру секции или ячейки) ставит в соответствие *“метку X”* либо *“пусто”*

ЛЕНТА МАШИНЫ ПОСТА

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
5. Машина Поста (2)

Программа машины Э. Поста – непустой список, обладающий свойствами:

- последовательность команд занимает сплошной массив;
- все адреса программных ячеек действительны.

Формат команды:

<Номер (адрес) команды>. <КОП> <Адрес следующей команды>

где КОП – Код ОПерации.

Виды команд:

$PC.\Rightarrow PC_1$;

$PC.\ \mathbf{X}\ PC_1$;

$PC.\Leftarrow PC_1$;

$PC.\ \text{§}\ PC_1$;

$PC.\ \mathbf{?} \to PC1 \to PC2$.

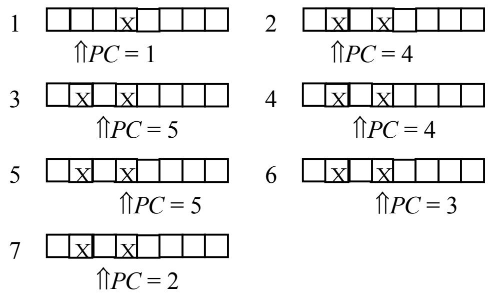
PC – Program Counter.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
5. Машина Поста (3)

Пример программы

1. **X** 4
2. § 3
3. $\Leftarrow$ 2
4. $\Rightarrow$ 5
5. **?** $^{\to 4}_{\to 3}$

Лента машины Поста



ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
5. Машина Поста (4)

Непрерывная последовательность отмеченных ячеек, по Посту, называется **массивом**.

Числа $n = \{0, 1, 2, \ldots\}$ в машине Поста представляется массивом ячеек длиной $n + 1$.

	X	X	X				X	X	X	X	X	X				X			X	X		

$n = 2$ $n = 5$ $n = 0$ $n = 1$

Замечание: В зависимости от фактического положения головки “считывания /записи” (RW) возможны различные вари анты программ обработки массивов.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
6. Инкремент числа (1)

Задача прибавления единицы к числу (инкремент числа) формулируется так.

Необходимо написать программу машины Поста, которая для любого значения *n*, будучи применена к произвольному состоянию класса A_n, даёт в состояние класса E_{n+1} результативную остановку.

Классы A_n определяются как местоположения головки считывания /записи относительно массива, а классы E_{n+1} – как результат инкремента.

Пусть позиция головки считывания/записи находится

- напротив какой-либо ячейки массива либо
- левее массива.

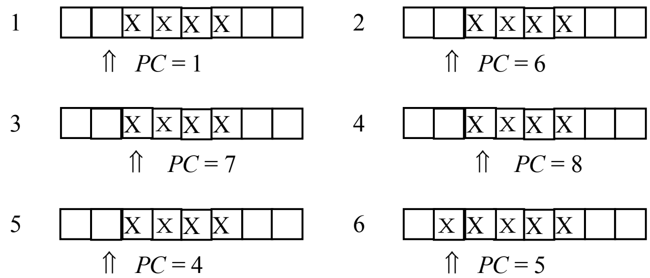
ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
6. Инкремент числа (2)

ТЕКСТ ПРОГРАММЫ

1. **?** $^{\to 6}_{\to 2}$
2. $\Leftarrow$ 3
3. **?** $^{\to 4}_{\to 2}$
4. **X** 5
5. СТОП
6. $\Rightarrow$ 7
7. **?** $^{\to 6}_{\to 8}$
8. $\Leftarrow$ 4

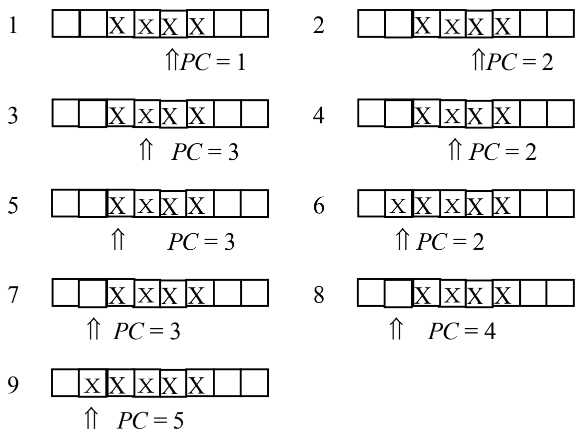
ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
6. Инкремент числа (3)

СЛУЧАЙ ПЕРВЫЙ



ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
6. Инкремент числа (4)

СЛУЧАЙ ВТОРОЙ



ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
7. Формулировка Гипотезы Поста

Пусть имеем два алгоритма с совпадающими совокупностями исходных данных.

Алгоритмы 1 и 2 называются **равносильными**, если **к любому** данному из общей их совокупности они либо **оба применимы**, либо **оба не применимы**, а если применимы, то **дают одинаковый результат**.

Гипотеза

Каждый алгоритм, все результаты которого суть числа, а областью исходных данных служат N, N^k или N^∞ равносильны алгоритму с такой же совокупностью возможных исходных данных, задаваемой некоторой программой машины Поста.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
8. Вычислимые функции и тезис Поста (1)

Вычислимая функция – это функция, которая вычислима.

$$\mathbf{A} \longrightarrow \mathbf{B}$$

Область отправления — Область прибытия

A: N, N^k и N^∞ — **B**: N, N^k и N^∞

Пример: $N^2 \to N.$ $\quad < x, y > \to \begin{cases} x - y, & x \ge y; \\ \text{undefined}, & y > x. \end{cases}$

Пусть существует некий алгоритм A:

$$f(x_1, x_2, ..., x_n) = \begin{cases} \text{результат применения } A \text{ к кортежу } < x_1, x_2, ..., x_n >; \\ \text{не определено, если } A \text{ не применим к кортежу } < x_1, x_2, ..., x_n > . \end{cases}$$

Функция f из N^k в N или из N^∞ в N *называется вычислимой на машине Поста* (вычислимой по Посту), если существует программа P, вычисляющая функцию f для машины Поста, обладающая свойствами:

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
7. Вычислимые функции и тезис Поста (2)

1. если $f(x_1, x_2, \ldots, x_k) = y$, то программа P, применяемая к исходным данным $<x_1, x_2, \ldots, x_k>$, заканчивает свою работу, после чего на ленте *остаётся запись числа* y;
2. если значение $f(x_1, x_2, \ldots, x_k)$ не определено, то применение программы P к исходным данным $<x_1, x_2, \ldots, x_k>$ *не приводит* к результативному останову.

Пусть даны:

- некоторый класс исходных данных N , N^k ($k > 1$) или N^∞, который обозначим через E и
- функция f из E в N.

Имеем пару задач:

(П) составить программу машины Поста, вычисляющую функцию f ;

(А) составить алгоритм, вычисляющий функцию f .

Тезис Поста.

Пусть E – одно из множеств N , N^k ($k > 1$) или N^∞. Тогда **класс** вычислимых функций из E в N **совпадает** с классом вычислимых на машине Поста функций из E в N.

ВЫЧИСЛИТЕЛЬНАЯ МОДЕЛЬ (МАШИНА) ПОСТА.
9. Постулат Поста

УТВЕРЖДЕНИЕ ИЛИ ПОСТУЛАТ ПОСТА

Задача на составление программы, приводящей от исходного данного к результирующему числу (и не приводящей ни к какому результату, если результирующего числа не существует), тогда и только тогда имеет решение, когда имеется какой-нибудь общий способ, позволяющий по произвольному исходному данному выписать результирующее число и не выдающий никакого – заведомо ложного – в этом случае результата, коль скоро результирующего числа не существует.