

Лекция 12

Random Access Machine (RAM). Машина Нинского.

Нормальные алгоритмы Маркова.

УТВ (Теория Тьюринга) Алгоритм может быть реализован в виде МТ.

Но если, вычисление процедурным способом:

1) Random Access Machine (RAM)

2) Машина Нинского

3) Нормальные алгоритмы Маркова

Также есть функциональные определения алгоритма.

Random Access Machine (RAM).

Устройство:

- 1) Возможно обращаться к ячейкам памяти по их индексу.
- 2) Одна ячейка хранит целое число
- 3) Ячейки разделяются на 3 группы: вход, память и выход. В каждой группе ячейки закодированы целыми неотрицательными числами
- 4) Кроме ячеек RAM включает в себя программу — список команд.

Команды:

<наименование команды> <так операнда> <операнд>

- 1) Арифметические операции (второй операнд берется из ячейки номер 0)

2) Текущая с выходной ленты

3) Запись на выходную ленту

Операнды:

Операнд - целое число. Типы операндов:

1) Константы (=)

Значение $v(i)$ константы i совпадает с ней самой

2) Переменные (λ)

Значение $v(i)$ переменной i равно числу, записанному в ячейке с номером i .

3) Указатели (*)

Значение $v(i)$ указателя i равно числу, записанному в ячейке с номером, который содержится в строке i .

Работа:

Начинается с исполнения первой команды.

Все команды делятся на 2 типа: Операции и переходы.

После исполнения операции происходит выполнение следующей в списке команд.

Исполнение перехода определяет номер команды, которая должна исполняться следующей.

Работа заканчивается в одном из следующих случаев:

1) Исполнена последняя команда, и это операция.

2) Исполнена программа перехода, но получившее в результате её число не является номером команды.

3) Очередная операция не может быть исполнена.

В начале работы ячейки RAM пусты кроме нескольких первых входных.

Состояние RAM - это гетверка $(n; x; r; y)$, где n - положительное целое число, список команд, x, r, y - бесконечные последовательности целых чисел, задающие состояния входа, памяти и выхода.

Команда задаёт отображение на множество состояний: $(n; x; r; y) \rightarrow (n'; x'; r'; y')$. Ко-

манды-операции: (для них всегда $n' = n + 1$)

1) store u - записывает $r(0)$ в ячейку с номером $r(u)$, то есть
 $r'(v(u)) = r(0)$

2) load u - записывает в ячейку содержимое ячейки с номером $v(u)$,

$$\text{то есть } r'(0) = r(v(u))$$

3) add u - сложение $r'(0) = r(0) + r(u)$

4) sub u - вычитание $r'(0) = r(0) - r(u)$

5) mul u - умножение $r'(0) = r(0) \cdot r(u)$

6) div u - целочисленное деление $r'(0) = r(0) / r(u)$

7) read u - чтение $r'(0) = x(v(u))$

8) write u - запись $y(v(u)) = r(0)$

Команды - переходы:

- 1) goto u - переход к команде с номером $v(u)$, то есть $n' = v(u)$
- 2) ifzero u - если $r(0) = 0$, то $n' = v(u)$, иначе $n' = n+1$
- 3) ifpos u - если $r(0) > 0$, то $n' = v(u)$, иначе $n' = n+1$

RAM - это конечная последовательность команд описанного выше языка.

Число регистров RAM счётно.

Множество Марковых:

Конечное число регистров R_1, R_2, \dots, R_n , каждый из которых содержит целое незнакомое число.

Обозначим $r(i)$ содержимое регистра с номером i .

Три типа команд:

- 1) inc i, j - увеличить значение регистра i на 1 и передать управление команде с номером j .
- 2) dec i, j, k - если значение регистра i равно 0, то передать управление команде с номером j , иначе уменьшить на 1 значение регистра i и передать управление команде с номером k .
- 3) halt - остановиться

В начальной конфигурации значение всех регистров, кроме первого, равно 0.

Результат работы - значение первого регистра после остановки.

Пример: Обнуление регистра

1) dec i, 2,

2) halt

Пример: Копирование регистра:

1) dec i, 1, 2

2) inc j, 3

3) inc k, 1

4) halt

Пример: Проверка четности

1) dec i, n, 2 }
2) dec i, m, 1 }

- уменьшает значение регистра i до 0 и передаёт управление
команде с номером n , если в начале работы регистр i
содержал чётное число, m - если нечётное.

Пусть A - конечный алфавит.

Оп Нормальным автоморфизмом (алгоритмом) (Маркова) в алфавите A называется
конечная последовательность правил подстановки, имеющих вид $u \rightarrow v$ или
 $u \rightarrow uv$, где слова $u, v \in A^*$. Правила вида $u \rightarrow v$ называются заключительными

Оп Нормальным алгоритмом над алфавитом A называется такой ^{нормальный} алгоритм

в некотором алфавите B таком, что $A \subset B$

Оп Последовательность правил подстановки называется схемой алгоритма.

Применение правила подстановки заканчивается в замене самого левого ~~внешнего~~
под слова и на под слово v . Если такого под слова нет, то говорят, что
рассматриваемое слово не поддаётся данному правилу.
На каждом шаге применяется первое из правил схемы, применимое к данному
слову.

Алгоритм заканчивает работу, когда слово не поддаётся схеме или после применения
заключительного правила.