

Vistula, IT Faculty, 2014

Algorithms and Complexity

Dmitry A. Zaitsev

<http://daze.ho.ua>

Lecture 3:

Turing machine as a model of computations

Turing machine components

...	λ	λ	a	a	b	a	b	b	λ	λ	...
-----	-----------	-----------	---	---	---	---	---	---	-----------	-----------	-----

^

Tape –
infinite to both sides
array of cells



λ - an empty (blank) symbol initially written in all the cells

Input – a word written on the tape before TM work

Output – a word read from the tape when TM halts

A working zone – a minimal part of the tape containing nonempty symbols

Control head – a current state q and a program of TM work

Configuration of TM – $\alpha q x \beta$

Rules of TM work

At a step:

- Control head stays in state q and reads the current cell symbol x
- Depending on the pair (q,x) it
- Writes a new symbol x' into the current cell
- Moves one cell left (L), or one cell right (R), or stays still (S)
- Goes to the next state q'

TM example

Addition of natural numbers in unary numeral system

...	λ	1	1	1	+	1	1	1	1	λ	...
-----	-----------	---	---	---	---	---	---	---	---	-----------	-----

q_0



...	λ	λ	1	1	1	1	1	1	1	λ	...
-----	-----------	-----------	---	---	---	---	---	---	---	-----------	-----

q_f

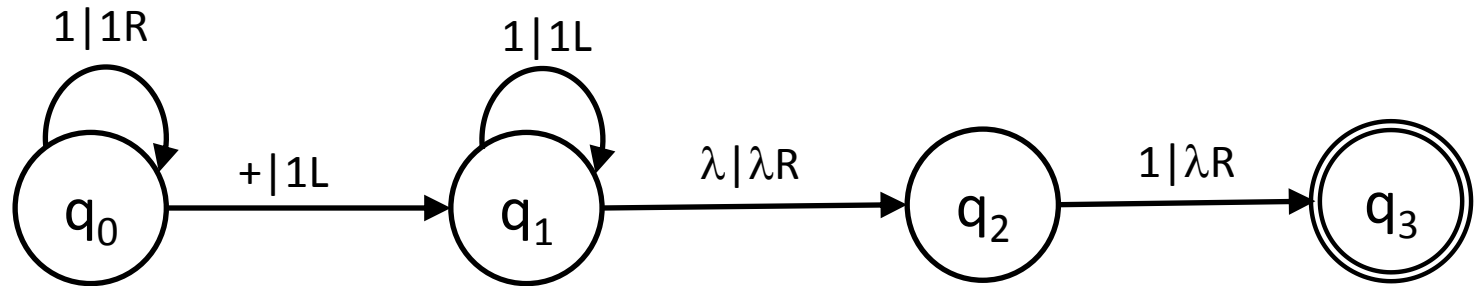
$$3 + 4 = 7$$

$$q_0 111 + 1111 \Rightarrow q_f 1111111$$

Algorithm:

- Go right to symbol “+”
- Replace “+” by “1”
- Go left to symbol “ λ ”
- Go one cell right
- Replace “1” by “ λ ”
- Halt

TM example

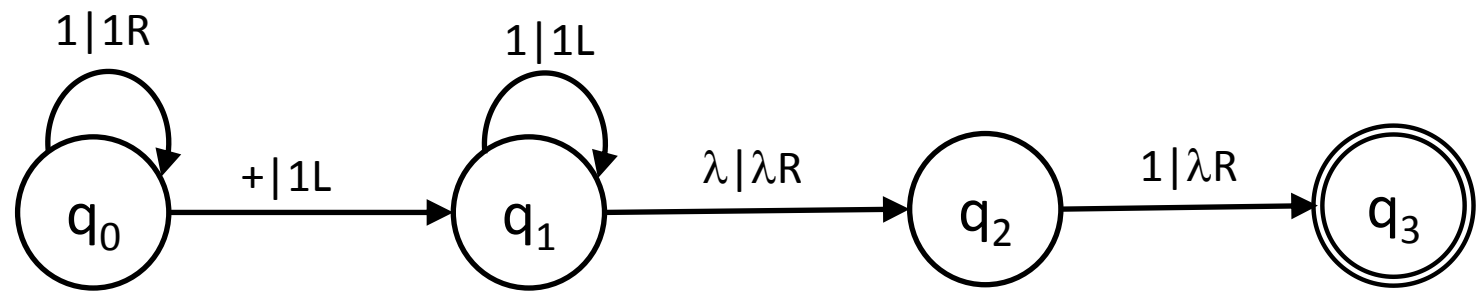


Tracing TM work for $3 + 4 = 7$

$\lambda q_0 111+1111\lambda \rightarrow \lambda 1 q_0 11+1111\lambda \rightarrow \lambda 11 q_0 1+1111\lambda \rightarrow$
 $\lambda 111 q_0+1111\lambda \rightarrow \lambda 11 q_1 111111\lambda \rightarrow \lambda 1 q_1 1111111\lambda \rightarrow$
 $\lambda q_1 11111111\lambda \rightarrow q_1 \lambda 11111111\lambda \rightarrow \lambda q_2 11111111\lambda \rightarrow$
 $\lambda \lambda q_3 11111111\lambda$

TM representation

Graph – state diagram



Tabular

State \ Symbol	1	+	λ
q ₀	1 R q ₀	1 L q ₁	-
q ₁	1 L q ₁	-	λ R q ₂
q ₂	λ R q ₃	-	-
q ₃	-	-	-

Program
(q, x, x', v, q')

(q₀, 1, 1, R, q₀)
 (q₀, +, 1, L, q₁)
 (q₁, 1, 1, L, q₁)
 (q₁, λ, λ, R, q₂)
 (q₂, 1, λ, R, q₂)

TM formal definition

$$M = (X, Q, \delta, q_0, q_f)$$

$X = \{x_0, \dots, x_{n-1}\}$ — a finite alphabet of tape symbols

$Q = \{q_0, \dots, q_f\}$ — a finite alphabet of internal states

$T: Q \times X \rightarrow X \times V \times Q$ — a transition function

$V = \{L, S, R\}$ — an alphabet of control head moves

Variants of definition:

- A set of final states
- Only left and right moves
- No final state – halt when there is no instruction

Thesis of Church

Any algorithm can be implemented as a corresponding Turing machine.

Intuitive concept of algorithm and concept of Turing machine coincide.

Other examples of TM: arithmetic in binary numbering system, reversing words etc

Programming in TM – composition of TM: sequential, parallel, branching, loops

Variants of TM:

- Tape infinite to one side
- A few tapes (multitape)
- Weak – infinite repetition of blank words

Universal TM

Any TM can be represented as a binary TM with $X=\{0,1\}$

Any TM can be encoded as a sequence of binary numbers $E(M)$

The simplest encoding:

A TM program – l_1, l_2, \dots, l_N

x, q, x', v, q' are encoded in unary numeral system and separated with “0”

Instructions l_k are separated with “00”

An input word and a program are separated with “000”

A universal TM accepts, as an input, word $E(w)E(M)$ and produces the word $E(M(w))$.

$E(w)$ represents input data and $E(M)$ represents a program of its work.

A universal TM exists –

the smallest examples T.Neary&D.Woods of 6 states and 4 symbols