Universal Turing Machine

Prof. (Dr.) K.R. Chowdhary
Email: kr.chowdhary@gmail.com

Formerly at department of Computer Science and Engineering
MBM Engineering College, Jodhpur

Thursday 13th November, 2025

K.R. Chowdhary

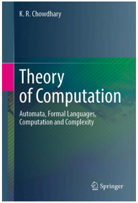
Theory of Computation

Automata, Formal Languages, Computation and Complexity

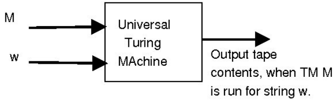
Focuses on pedagogy in its writing, that represents a refreshing approach

Ensures comprehensive and enjoyable learning

Undergone a rigorous classroom testing

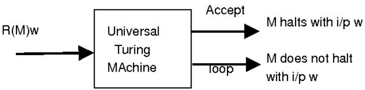


- A 3-tape TM, 2D-TM, and NDTM can be simulated by a standatd TM. Also, A TM can be also simulated by a TM.
- Let Input $= [M, w]$ to a TM M'. Output of M' is what, when M runs with input w. M' is Universal Turing machine (UTM).
- A UTM can be designed to simulate the computations of an arbitrary TM M. To do so, input to UTM must contain representation of both - machine M and input w to be processed by M.



TM Simulation on another TM

- Let there is TM M that accepts by halting. The UTM M' for this is:
  with Input string $= R(M)w$, where $R(M)$ is representation of M.
- Output-1: Accept (indicates that M halts with input w), output-2: loops, i.e., M does not halt with input w, i.e. computation of M does not terminate.
- The machine M' is called universal TM, as computation of any Turing machine can be simulated by M'.



Design a string representation of a TM M

Because of the ability to encode arbitrary symbols as strings over $\{0, 1\}$, we consider Turing machine with inputs $\{0, 1\}$ and tape symbols $\Gamma = \{0, 1, B\}$

Encoding of elements of M:

Symbol	Encoding
0	1
1	11
B	111
q_0	1
q_1	11
...	...
q_n	1^{n+1}
L	1
R	11

- The states of M are assumed to be $\{q_0, q_1, \ldots, q_n\}$. TM M is defined by its transition function:

$$\delta(q_i, a) = (q_j, b, d)$$

where,
$q_i, q_j \in Q; a, b \in \Gamma; d \in \{L, R\}$

- Let $en(z)$ denote the encoding of z. Thus, transition $\delta(q_i, a) = (q_j, b, d)$ is encoded by string:
$en(q_i)0en(a)0en(q_j)0en(b)0en(d)$.
The symbol 0 separates the different components of δ.

Encoding of elements of M

Representation of machine M is constructed from encoded transitions. Two consecutive 0s separate transitions. Beginning and end of complete representation are defined by three 0s.

Consider the Transitions:

Transition	Encoding
$\delta(q_0, B) = (q_1, B, R)$	101110110111011
$\delta(q_1, 0) = (q_0, 0, L)$	1101010101
$\delta(q_1, 1) = (q_2, 1, R)$	110110111011011
$\delta(q_2, 1) = (q_0, 1, L)$	1110110101101

- The machine M is represented by string: 000101110110111011 0011010101010011011011101101100111011010110100 0

  Simulation of M on Universal TM M'

  Verification of representation of M: TM can be constructed to check whether an arbitrary string $u \in \{0, 1\}^*$ is encoding of deterministic TM M. Computations examines whether 000 is prefix, followed by finite sequences of encoded transitions are separated by 00s, then finally 000.
- M is deterministic if $Q \times \Gamma$ in every encoded transition is unique.

Simulation of TM M on 3-tape DTM M'

- Tape-1 holds $R(M)w$. Tape-3 simulates computations of of M for input w. Tape-2 acts as working tape.
- If input u is not of the form $R(M)w$ for deterministic TM M and string w on tape-1, the M' moves to right forever.

1. w is copied from tape-1 to 3, with tape head at begin of w. $\therefore$ tape-3 is initial configuration of M with input w.
2. Encoding of q_0, i.e., 1 is written tape-2. (for future steps, we call it q_j).
3. Transition of M is simulated on tape-3. The next transition is determined by symbol scanned on tape-3 and state encoded on tape-2. Let these are a and q_i.
4. Tape-1 is scanned for a and q_i as first two components of a transition. If not found, M' halts by rejecting input.
5. If tape-1 consists the encoded information for above, i.e., $\delta(q_i, a) = (a_j, b, d)$, then
   (a) q_i replaced by q_j on tape-2.
   (b) b is written on tape 3, and tape head on tape-3 is moved for direction given in d.
6. Go back to step 2, and carry on computation by simulating M.

Bibliography

Chowdhary, K.R. (2025). Extensions of Basic Turing Machine. In: Theory of Computation. Springer, Singapore. https://doi.org/10.1007/978-981-97-6234-7_10