

Computational Complexity of TMs

KR Chowdhary
Professor & Head
Email: kr.chowdhary@gmail.com

Department of Computer Science and Engineering
MBM Engineering College, Jodhpur

December 7, 2012

- Complexity is *Rate of growth* of computation in time and space.
- **DTM:** The time complexity is measured in terms of number of transitions in a computation, and space complexity is maximum number of tape squares used.
- **Multi-tape TM:** Complexity differs **polynomially** from standard TM
- Computations of a computer can be simulated on a TM, and it can be shown that no. of transitions in TM = polynomial expression \times no. of instructions executed in computer.
- \therefore , resource bounds in TM provide practical information about complexity of algorithms and computer programs.

Complexity:

Objective: 1) Assessment of Algorithms of problems, 2) Comparison of difficulty of problems.

Problem Examples:

1. Array sort: Input: $A_1 \dots A_n$, Output: $A_{1'} \dots A_{n'}$
2. Squaring a Matrix: Input: $B : n \times n$, output: $C = B^2$
3. Path problem for directed graph: Input: $G = (V, E)$, Output: Yes, if there is a path from v_i to v_j , else No.

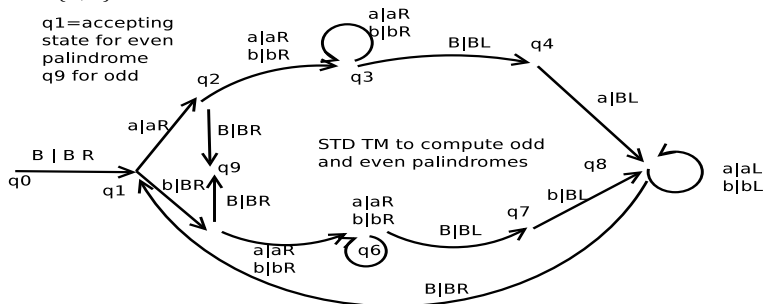
- 4 Acceptance by a TM that halts for all inputs: Input: String w , output: Yes, if M accepts w , No otherwise
- Case 1,2: Computes function, case 3,4: are decision problems. Resource usage is measured in: case 1 as no. of data movements, in case 2 as number of scalar multiplications, in 3 as No. of nodes visited, 4 as No. of transitions in a computation. There is no definite relation in complexities in above.
- Church-Turing thesis says that TM is an effective computing machine, \therefore **time** and **space** complexities of TMs describe **no. of transitions** and **& number of squares** on tape for all the problems.
- Rate of growth:

Let $f(n) = n^2$, and function $g(n) = n^3$.

$\therefore f \in O(g)$

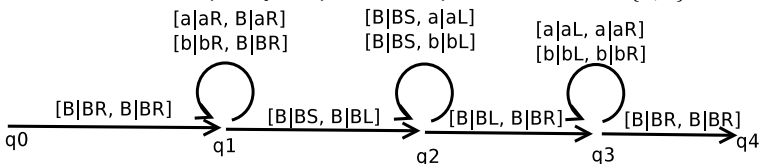
$g \notin O(f)$, $\therefore n^2 \in O(n^3)$, since $n^2 \leq n^3$

- TC (Time complexity) is measured in terms of number of transitions, for TM M for input $|w| = n$
- Example: Analyze the computation for palindrome problem over $\Sigma = \{a, b\}$ for TM



- Let $tc_m(n)$ is time complexity. Thus, $tc_m(0) = 1$, $tc_m(1) = 3$, $tc_m(2) = 6$, $tc_m(3) = 10$, $\therefore tc_m(n) = \sum_{i=1}^{n+1} i = (n+2)(n+1)/2 = O(n^2)$

- Exercise: Time complexity for palindrome problem over $\Sigma = \{a, b\}$



B is left end marker
and blank symbol

q_4 is accepting state

- Working: Loop1= Copy, Loop2=Rewind tape2, Loop3=Compare tape1 and tape2.
- Accepting computation: 3 passes, $tc_m(n) = 3(n+1) = 1$, is polynomial.
 \therefore complexity of standard TM = polynomial \times time complexity of 2-tape TM.

Theorem

Let L be language accepted by K -track DTM M with Time complexity $tc_M(n)$, then L is also accepted by standard TM M' with T.C. $tc_{M'}$, then $tc_M(n) = tc_{M'}(n)$.

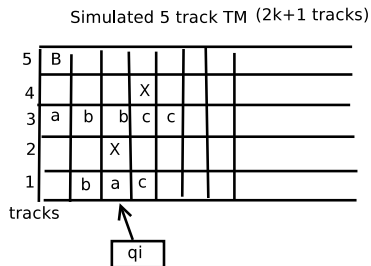
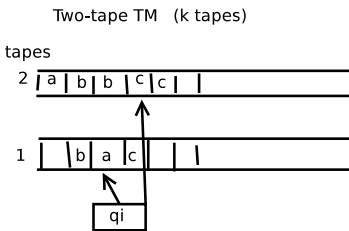
Proof.

- In k -track TM $\delta(q_i, x_1, \dots, x_k)$, where x_i is symbol in track i .
- The same transition in std-TM (one tape) is $\delta(q_i, [x_1, \dots, x_k])$, where k -tuple is single alphabet in std-TM M' . $\therefore, tc_M = tc_{M'}$.



Simulation of multi-tape TM by Multi-track TM

- K tape TM can be simulated by a $2k + 1$ track TM



- In Two tape: Two heads, Multi-track: one head. The Symbol X in multi-track indicates virtual head position of two tape TM. To simulate a transition of two-tape, the 5 track, the TM locates a symbol at corresponding virtual head (X), reads the symbol under that, and performs transition as that in two tape TM. Obviously, the complexity of multi-track is more than that of k -tape TM.

Theorem

Let L be language accepted by K -tape DTM M with Time complexity $tc_M(n) = f(n)$, then L is also accepted by standard TM N with time complexity $tc_N(n)$, where $tc_N(n) \in O(f(n)^2)$.

Proof.

- We will first simulate K -tape TM on multi-track TM M' . The no. of tracks will be $2k+1$, as proved earlier.
- \therefore , we should show only: $tc_{M'} \in (f(n)^2)$. (Because complexities of multi-track and standard TM are equal).
- Let us try to simulate the farthest t^{th} transition of M on M' .
 1. There are k -even tracks for each head position symbol X . There are $2 \times t$ transitions. Total transitions = $2(t+1)$, one extra for head position, on $2t+1^{\text{th}}$ track. Contd ...



2. Similarly, for reading-writing of symbols on all odd tracks, transitions are $2t$.

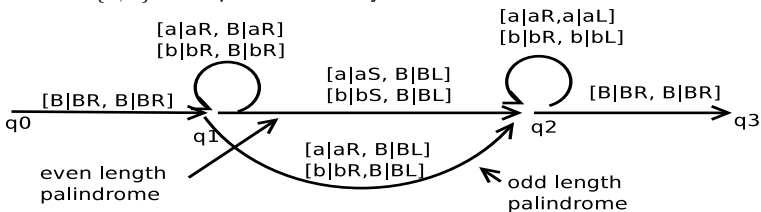
3. For k tape TM M , total transitions =
 $k(2t + 2(t + 1)) + 1 = 4kt + 2k + 1$. Thus
 $tc_{M'}(n) \leq 1 + \sum_{t=1}^{f(n)} (4kt + 2k + 1) \in (f(n)^2)$.

- Since, time complexities of Std TM and multi-track are same, the above time complexity is of std TM. \square

- A DTM generates and examines all possibilities
- A NDTM employs a Guess-and-check strategy, and needs to determine if one of the possibilities provide the solution
- **Example:** To find out if k is composite number, we try to divide it by all the possible factors of it. If one exists, then it is composite. The set of all the composite numbers is language COMPOSITS.
- A NDTM chooses a guess value, and a single division (verification) determines that it is factor.
- A string is accepted by NDTM if at least one of the computation terminates in accepting state.

Two-tape NDTM to decide a palindrome language

- Let $\Sigma = \{a, b\}$. The palindrome may be odd or even.



B is left end marker and blank symbol
, q3 is accepting state

- Operations:

1. Read and copy 1st half from tape1 to tape2,
 2. nondeterministically decide the centre, at tape 1,
 3. compare 2nd half on tape 1 with first half on tape 2.
- No. of transitions for input $|w| = n$, is $n+2$ for odd n , $N+3$ for even n .
 $tc_M(n) = O(n)$.

Theorem

Let L be language accepted by NDTM M with $tc_M(n) = f(n)$, for input $|w| = n$. Then, L is accepted by DTM M' with T.C. $tc_{M'} = O(f(n)c^{f(n)})$, where c is maximum number of transitions for any State \times Symbol pair of M .

Proof.

- Let $M = (Q, \Sigma, \Gamma, \delta, q_0, H)$, that halts for all the inputs. Transformation from M to M' is done as follows:
- A unique computation of M has sequence $(m_1, \dots, m_n), 1 \leq m_i \leq c; m_i$ is which of the c possible transitions are executed in i th step. For $|w| = n$, maximum transitions of M is $f(n)$. Contd next ...



- For one computation of M , the M'
 1. generates sequence m_1, \dots, m_n
 2. Simulate M for this sequence
 3. if solution not found, goto step a.
- In worst case no. of transitions in M' is $c^{f(n)}$, as so many sequences are to be analyzed.
- T.C. of M' is $O(f(n)c^{f(n)})$. \square

Note: This is similar to a binary tree where the lowest level there are 2^h nodes, where h is the responsibility of .