Turing Machines Extensions

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

Department of Computer Science and Engineering
MBM Engineering College, Jodhpur

December 7, 2012

Ways to Extend Turing Machines

Many variations have been proposed:

- Multiple tape Turing machine
- Multiple track Turing machine
- Two dimensional Turing machine
- Multidimensional Turing machine
- Two-way infinite tape
- Non-determinic Turing machine
- Combinations of the above
- Theorem: The operations of a TM allowing some or all the above extensions can be simulated by a standard TM.The extensions do not give us machines more powerful than the TM.
- The extensions are helpful in designing machines to solve particular problems.

Multiple Tape TM

Variants of TM:

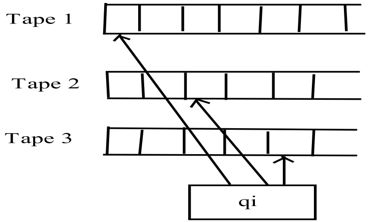
- For example, two tape Turing machine, each with its own read-write head, but the state is common for all.
- In each step (transitions), TM reads symbols scanned by all heads, depending on those and current state, each head writes, moves R or L, and control-unit enter into new state.
- Actions of heads are independent of each other
- Tape position in two tapes: $[x, y]$, x in first tape, and y in second, and δ is given by:

  $\delta(q_i, [x, y]) = (q_j, [z, w], d, d), \quad d \in \{L, R\}$

δ	a, a	B, a	a, B	B, B
q_0	q_1, a, b, L, R	q_2, b, B, L, L	$q_0, b, B, L, R,$	$\ldots$

- A standard TM is multi-tape TM with single tape. Hence, every Recursively enumerable language is accepted by multi-tpae TM
- Example: They are more suited for specific applications, e.g., copying string from one tape to another tape.

Multiple Tape TM



A transition in amulti-tape turing machine, for k number of tapes:

$\delta : (Q - H) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R\}^k$

$\delta(q_i, a_1, \ldots, a_k) = (q_j, (b_1, \ldots, b_k), (d_1, \ldots, d_k))$

The steps to carry out a transition are:

1. change to next state;
2. write one symbols on each tape;
3. independently reposition each tape heads.

Multiple Tape TM

Simulation of a three-tape TM by standard TM:

- Let the contents of a three tape TM M are:

  0**1**010B Tape 1; bold character is R/W head position

  aa**a**B Tape 2; ...

  baB Tape 3; ...

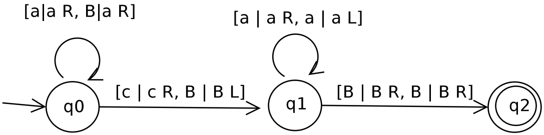
- Tape contents on simulated standard TM S:

  0**1**010#aa**a**#**b**aB , # is separator for contents of 3 tapes.

- In practice, S leaves an extra blank before each symbol to record position of read-write heads
- S reads the symbols under the virtual heads (L to R).
- Then S makes a second pass to update the tapes according to the way the M's transition function dictates.
- If, at any point S moves one of the virtual heads to the right of #, it implies that head moved to unread blank portion of that tape. So S writes a blank symbol in the right most of that tape.Then continues to simulate.

  $\Rightarrow$ control will need a lot more states.

Two-tape TM to recognize language $a^i ca^i$



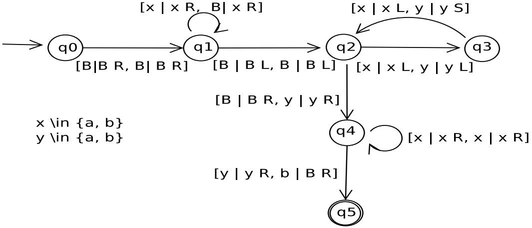
The above transition disgram shows a two-tape TM, recognizing the language: $L = \{a^i ca^i\}$.

Working:

1. The original string is on tape 1
2. Copies the string w on tape2, until middle c is reached. Now the head 1 moves to right and head 2 moves to left, comparing 2nd half of tape1 with 1st half on tape 2, and this comprision is done in reverse order. Note that it can also recognize language wcw^R.
3. Since, it always halts, the langauge is **recursive**. And, the recisive language always provides a **decision** procedure for membership.

Multi-Tape TM

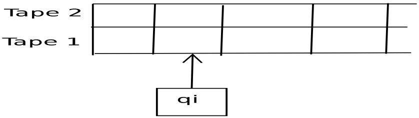
Example: Construct 2-tape TM to recognize the language
$L = \{ww | w \in \{a, b\}^*\}$.



steps:(Note: $x \in \{a, b\}$, $y \in \{a, b\}$)

1. Initially the string *ww* is on tape-1. Copy it to tape-2, at the end both R/W heads are at right most.
2. *Move both heads left:* Head-1, 2-steps and head-2, 1-step each time.
3. *Move both heads right*, each one step. Head-1 moves in first *w* of *ww* and head-2 moves in second *w*, comparing these *w* in q_4. In $q_3 \rightarrow q_4$ transition, the move '$y|y\ S$' keeps head2 *stationary*.

Multiple track TM



- The tape is divided into tracks. A tape position in n-track tape contains n symbols from tape alphabets.
- Tape position in two-track is represented by $[x, y]$, where x is symbol in track 1 and y is in tack-2. The states, Σ, Γ, q_0, F of a two-track machine are same as for standard machine.
- A transition of a two-track machine reads and writes the entire position on all tracks.
- δ is: $\delta(q_i, [x, y]) = [q_j, [z, w], d]$, where $d \in \{L, R\}$. The input for two-track is put at track-1, and all positions on track-2 is initially blank. The acceptance in multi-track is by final state.
- Languages accepted by two-track machines are Recursively Enumerable languages.

Theorem

A language is accepted by a two-track TM if and only if it is accepted by a standard TM.

Proof.

- *Part 1:If L is accepted by standard TM then it is accepted by two-track TM also(simply ignore 2nd track), i.e.,*$[a, B]$.

  Part 2:
- *Let* $M = (Q, \Sigma, \Gamma, \delta, q_0, H)$ *be two track. Find one equivalent standard TM?*
- *Create ordered pair* $[x, y]$ *on single tape machine* M'.
- $M' = (Q, \Sigma \times \{B\}, \Gamma \times \Gamma, \delta', q_0, F)$ *with* δ' *as* $\delta'(q_i, [x, y]) = \delta(q_i, [x, y])$.

□

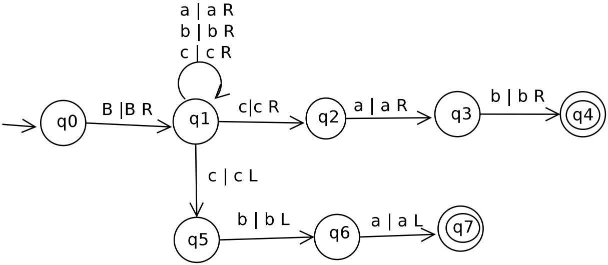
Nondeterministic TM (NDTM)

- Has finite number of choices of moves; components are same as standard TM; may have > 1 move with same input ($Q \times \Sigma$). Nondeterminism is like FA and PDA.
- A ND machine accepts by halting if there is at least one computation that halts normally when run with input w.
- Example: Find if a graph has a connected subgraph of k nodes (no efficient algorithm exists).Non-exhaustive based solution is Guess & check.
  1. NDTM: Arbitrarily choose a move when more than one possibility exists for $\delta(q_i, a)$.
  2. Accept the input if there is at least one computation that leads to accepting state (however, the converse is irrelevant).
- To find a NDTM for ww input, $w \in \Sigma^*$, you need to guess the mid point. A *NDTM* may specify any number of transitions for a given configuration, i.e.

  $\delta : (Q - H) \times \Gamma \rightarrow subset\ \ of\ \ Q \times \Gamma \times \{L, R\}$

NDTM to accept *ab* preceded/followed with *c*

Example: Example: $w = ucv$, where c is preceded by or followed by ab



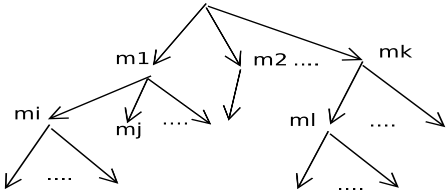
Approach: Read input a, b, c and write a, b, c respectively, and move R in each, at start state. Then with input c, Nondeterministically decide c, a, b by moving R in three states transitions or decide c, b, a by moving L in three other states transitions (i.e., abc)

Simulation of NDTM on Standard TM

- To accomplish the transformation of a NDTM to a determinsitic TM, we show that multiple computatiosn of a single input string can be sequentially generated and examined.
- A NDTM produces multiple computations for a single string. We show that multiple computations $m_1, \ldots, m_i, \ldots, m_k$ for a single input string can be sequentially generated and applied.(A computation m_i is $\delta(q_i, a) = [q_j, b, d]$, where $d \in \{L, R\}$.
- These computations can be systematically produced by adding the alternative transitions for each $Q \times \Sigma$ pair. Each m_i has $1 - n$ number of transitions. If $\delta(q_i, x) = \phi$, the TM halts.
- Using the ability to sequentially produce the computations, a $NDTM \;\; M$ can be simulated by a 3-tape TM $M^{'}$.
- **Every nondeterministic TM has an equivalent 3-tape Turing machine, which, in turn, has an equivalent standard TM(1-tape Turing machine).**

Simulation of NDTM by 3-tape TM:

- Approach: A NDTM may have more than one transition for same input (state × input symbol) pair. We may call these configurations as $c_0, c_1, \ldots, c_m$. These can be taken as child nodes generated from start node. Then, each child further generates some nodes as seen in fingure below.



A deterministic tree produced
for all possible transitions in NDTM

- Next, the tree created cna be searched in DFS or BFS, until accepting/halting state is reached. DBF is nto preferred because, it may go infinity. Therefore, the generated states are searched in BFS.
- Seaarch order is $m_1, m_2, \ldots, m_k, m_i, m_j, \ldots, m_l, \ldots,$

Simulation of a NDTM by 3-tape TM

- Tape-1 stores the input string, tape-2 simulates the tape of M, and tape-3 holds sequence $m_1, \ldots, m_i, \ldots, m_k$ to guide the simulation.
- Computation of $M^{'}$ consists following:
  1. A sequence of inputs $m_1, \ldots, m_i, \ldots, m_k$, where each $i, (i = 1, n)$ is written on tape-3.
  2. Input string is copied on tape-2.
  3. Computation of M defined by sequence on tape-3 is simulated on tape-2.
  4. If simulation halts prier to executing k transitions, computations of $M^{'}$ halts and accepts input, else
  5. the Next sequence is generated on tape-3 and computation continues on tape-2.

Two-way infinite tape

- There is single tape which extends from $-\infty$ *to* $+\infty$. One R-W head, $M = (Q, \Sigma, \delta, q_0, F)$

  . . . -3 -2 -1 0 1 2 3 . . ., is square sequence on TM, with R-W head at 0

  This can be simulated by a two-tape TM:

- $M' = (Q' \cup \{q_s, q_t\}) \times \{U, D\}$, where U = up tape head, D = down tape head, $\Sigma' = \Sigma, \Gamma' = \Gamma \cup \{B\}$, and

  $F' = \{[q_i, U], [q_i, D] | q_i \in F\}$. Initial state of M' is pair $[q_s, D]$. A transition from this writes B in U tape at left most position. Transition from $[q_t, D]$ returns the tape head to its original position to begin simulation of M.

Multi-Dimensional Tape:

- Single R-W head, but multiple tapes exists. Let the Dimensions be 2D. For each input symbol and state, this writes a symbols at current head position, moves to a new state, and R-W head moves to left or right or up or down.

  Simulate it on 2-tape TM:

- copy each row of 2-D tape on 2nd tape of 2-tape TM. When 2D TM moves head L or R, move the head on 2nd-tape of two-tape also L or R. When 2D head moves up, 2nd tape of two-tape scans left until it finds $*$. As it scans, it writes the symbols on tape-1. Then scans and puts remaining symbols on tape-1. Now it simulates this row (on tape-1).