Turing Machines

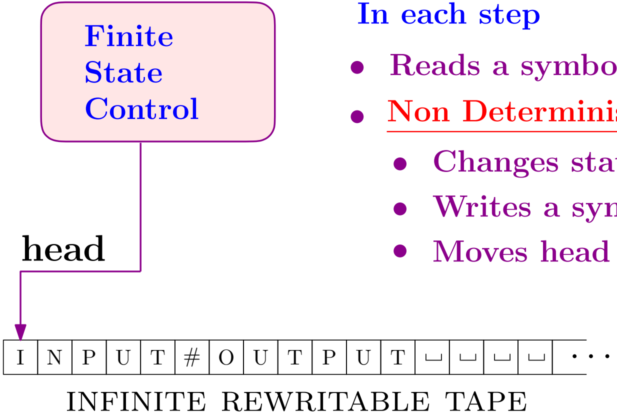
- Turing Machine: Model of Computation
- Turing Machine: Anatomy and Working
- Turing Machine: Formal Definition and Rules of Computation
- Recognizable and Decidable Languages
- Turing Machine: Levels of Abstraction
- Varaints of Turing Machine and The Church-Turing Thesis
- Non-Deterministic Turing Machine

Imdad ullah Khan

NonDeterministic Turing Machines

NonDeterministic Turing Machine

A NonDeterministic Turing Machine makes nondeterministic choices

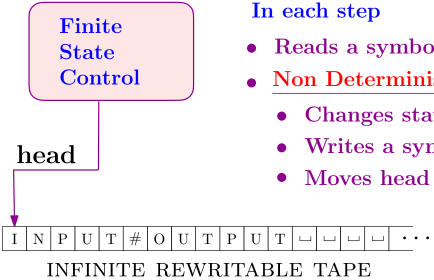

Finite State Control
head
I | N | P | U | T | # | O | U | T | P | U | T | ␣ | ␣ | ␣ | ␣ | ⋯
INFINITE REWRITABLE TAPE

In each step

- **Reads a symbol at the head**
- **Non Deterministically**
  - **Changes state**
  - **Writes a symbol at the head**
  - **Moves head to left or right**

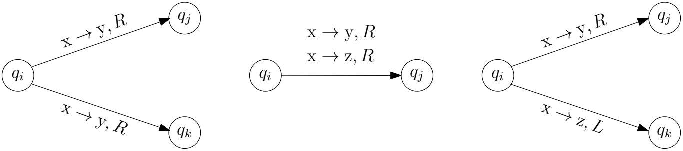
NonDeterministic Turing Machine

A NonDeterministic Turing Machine makes nondeterministic choices



In each step

- **Reads a symbol at the head**
- **Non Deterministically**
  - **Changes state**
  - **Writes a symbol at the head**
  - **Moves head to left or right**

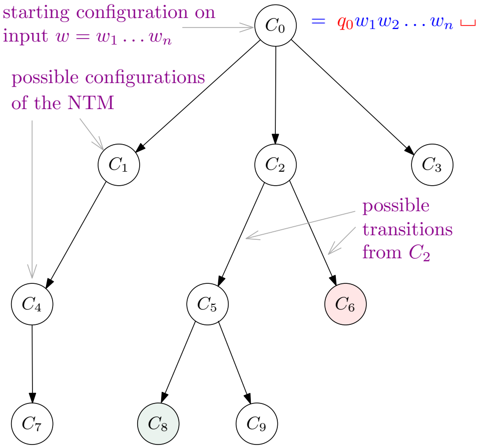


$$\delta : Q \times \Gamma \mapsto \mathcal{P}\left(Q \times \Gamma \times \{L, R\}\right)$$

NonDeterministic Turing Machine

For an NTM a computation is a tree of configurations reachable from the root (starting configuration $qw\sqcup$).

▷ For TM a computation is a sequence (path) of configurations

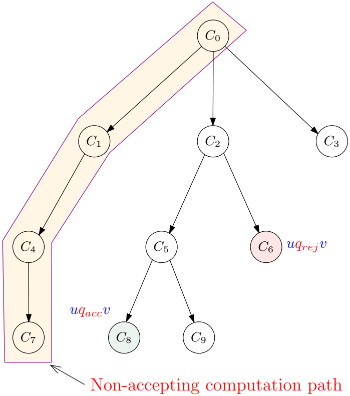


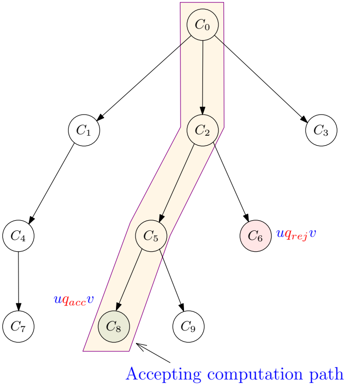
starting configuration on input $w = w_1 \dots w_n$
C_0
$= q_0 w_1 w_2 \dots w_n \sqcup$
possible configurations of the NTM
C_1
C_2
C_3
possible transitions from C_2
C_4
C_5
C_6
C_7
C_8
C_9

NonDeterministic Turing Machine

An NTM accepts a string *w* iff some computation path ends in an accepting configuration

i.e. if there is at least one sequence of configurations from the starting configuration to an accepting configuration





Turing Machine Variants: 2-tape-DTM = NTM

A Nondeterministic TM has equal power as a deterministic TM

A 2-tape deterministic TM D can simulate any NTM N

On a string w, using knowledge of N's finite control, D "traverses" the computation tree of N starting from $q_0 w$. If D encounters a configuration with q_{accept} D accepts w and stops simulation

What order should D traverse the computational tree of N on w?

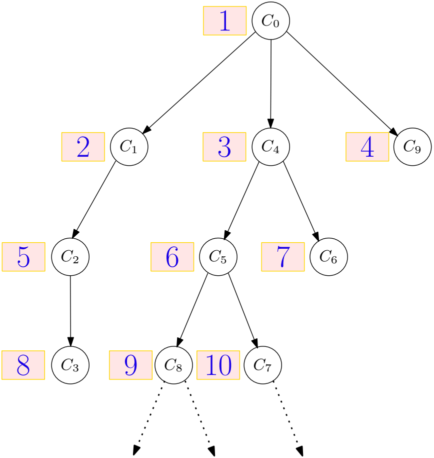
Will depth first search work?

What if N's computation tree has an infinite branch?

Recall how operating system schedule processes on CPU. What if a buggy process is scheduled on CPU?

Turing Machine Variants: 2-tape-DTM = NTM

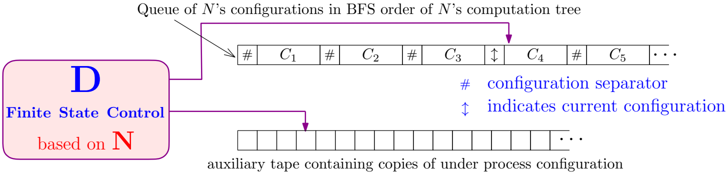
A 2-tape deterministic TM D can simulate any NTM N



- D "processes" N's configuration in a BFS fashion
- It maintains a queue of N's configurations with the next to be processed at head
- "processing configuration C"
  - Accept if C is an accepting configuration, else
  - Add to queue all configurations that yield from C in one move

Turing Machine Variants: 2-tape-DTM = NTM

A 2-tape deterministic TM D can simulate any NTM N

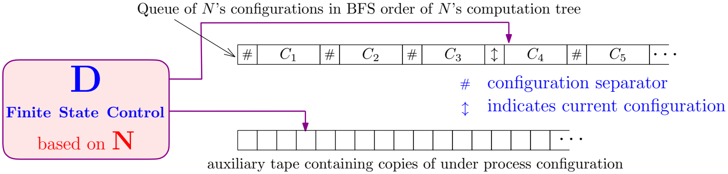


Queue of N's configurations in BFS order of N's computation tree
# | C_1 | # | C_2 | # | C_3 | ↕ | C_4 | # | C_5 | ⋯
D
Finite State Control
based on N
# configuration separator
↕ indicates current configuration
auxiliary tape containing copies of under process configuration

- D "processes" N's configuration in a BFS fashion
- Maintains a queue of N's configurations (in BFS order)
- "processing configuration C" (configuration preceded by symbol $\updownarrow$)
  - **Accept** if C is an accepting configuration (and stop simulation), else
  - If the state in C has t possible moves, D makes t copies of C in tape 2
  - On each copy simulate an N move, push resulting configuration on queue
  - Clear the auxiliary tape, mark the next configuration with $\updownarrow$
  - If no next configuration, then **reject** else repeat

Turing Machine Variants: 2-tape-DTM = NTM

A 2-tape deterministic TM D can simulate any NTM N


Queue of N's configurations in BFS order of N's computation tree
# | C_1 | # | C_2 | # | C_3 | ↕ | C_4 | # | C_5 | ⋯
# configuration separator
↕ indicates current configuration
D
Finite State Control
based on N
auxiliary tape containing copies of under process configuration

- Let t be the max number of non deterministic moves from a state
  ▷ maximum degree of a node in the digraph representation of N
- In k steps, N can reach $\leq 1 + t + t^2 + \ldots + t^k \leq kt^k$ configurations
- D's queue contain at most kt^k configurations ▷ exponential slowdown

Is this exponential slowdown necessary? essentially the P vs NP question