Nondeterministic TM I

- $\delta$: 
  \[ \delta : Q \times \Gamma \rightarrow P(Q \times \Gamma \times \{L, R\}) \]

  $P$: power set

- Example:
  \[
  q_0, a \rightarrow q_1, b, R \\
  \rightarrow q_2, c, L
  \]

- What if
  \[
  q_0, a \rightarrow q_{\text{accept}} \\
  \rightarrow q_{\text{reject}}
  \]
For NTM, $w$ accepted if one branch works
In this sense, unless all branches are finite
NTM $\rightarrow$ accept or endless loop
NTM: like an “acceptor”
Example of NTM I

- $A = \{ w \mid w \text{ contains } aab \}$
- State diagram
Example of NTM II

\[ a, b \rightarrow a, b, R \]

- \( q_0 \) to \( q_1 \) on \( a \) \( \rightarrow R \)
- \( q_1 \) to \( q_2 \) on \( a \) \( \rightarrow R \)
- \( q_2 \) to \( q_a \) on \( b \) \( \rightarrow R \)
- \( qr \) to \( q_0 \) on \( \square \) \( \rightarrow R \)
- \( qr \) to \( q_1 \) on \( \square \) \( \rightarrow R \)
- \( qr \) to \( q_2 \) on \( \square \) \( \rightarrow R \)
- \( qr \) to \( qr \) on \( \square \) \( \rightarrow R \)
Example of NTM III

- You may recall that this is an NFA example discussed before
- Only the first node is nondeterministic
Example of NTM I

- \( L = \{0^n \mid n \text{ composite number}\} \)
- From p. 204 of Lewis and Papadimitriou
- Composite number: product of two natural numbers
- Procedure
  - Nondeterministically choose \( p \) and \( q \)
  - Sequentially try \( p \) from 1 to \( n \)
  - Check if \( n = pq \)
    - This can be done by the earlier example

\[ \{a^n b^p c^q \mid n = p \times q\} \]
Example of NTM II

- Question: details about “non-deterministically” choose $p$ and $q$?
- If we sequentially try all $(p, q)$ combinations, then looks like we have a deterministic setting?
- Our generation of $p$ and $q$ can be non-deterministic
- Say we do a copy operation to generate $p$ elements. The TM can have an $\epsilon$ link to stop at any time point
Nondeterministic TM $\equiv$ deterministic TM

- easy
  A deterministic TM is a nondeterministic TM
- more difficult
- Like NFA we use a tree for processing the input (# branches finite)
- To traverse a tree we can do
  depth-first search
  or
  breadth-first
Nondeterministic TM $\equiv$ deterministic TM

- If using depth-first search, one branch may lead to $\infty$ steps
  Then we cannot consider other branches even if the input is accepted
- Thus we should consider breadth-first
- Fig 3.17: a deterministic TM to simulate a nondeterministic TM
Nondeterministic TM $\equiv$ deterministic TM

- Tape 1: input, never altered

Tape 1: 0 1 1 1 0 ...
Tape 2: x x 1 1 0 ...
Tape 3: 1 2 3 2 3 ...

CPU
Nondeterministic TM ≡ deterministic TM

- Tape 2: process one branch
- Tape 3: maintain the tree
- The key is the 3rd tape
- Suppose max ≠ branches 3
  At the 1st step: if contents of 3rd tape are 1 2
  ⇒ can go to 1 or 2 from \( q_0 \)
- The tree keeps growing. For example,
  1 2 12 13 2 12 13 21 22 23 121 123 13 21 22 23
Nondeterministic TM $\equiv$ deterministic TM

- What if 12 is a failed branch?
  - 12 13 21 22 23 12 131 132 21 22 23
- 12 fails, continue 131, no need to remove 12
Corollary 3.19 I

- **Definition**: NTM is a decider if all branches halt on all inputs.
- **Language decidable** $\iff$ some NTM decides it.
- $\Rightarrow$ easy, one TM decides it and TM is an NTM.
  This TM halts on all inputs (one branch).
- $\Leftarrow$:
  Now NTM terminates on all branches.
  We will construct a TM to accept the language.
  - each branch is finite
  - every input halts $\exists$ a finite max length.
Corollary 3.19 II

- \# branches finite at each node
- The tree to process this input is finite
- Write it in the 3rd tape
- We know a multi-tape TM is equivalent to a single-tape TM