Variants of TMs

1. TMs with stay-put
2. Multitape TMs
3. Nondeterministic TMs (NTMs)
4. TM with output (enumerators)
Breadth First Search

order of traversal:

queue
Nondeterministic TM

$$(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$$

$$\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\})$$
Nondeterministic TM

Each node corresponds to a configuration.
Nondeterministic TM: Accept
Nondeterministic TM: nonterminating branch
Theorems

Obs 1:
Every multitape TM has an equivalent single tape TM

Every nondeterministic TM has an equivalent deterministic TM
Obs 3: max number of children (= 3)

Obs 2:
Given: NTM N

Construct:
Multitape Deterministic TM D that simulates every branch of N
Idea: traverse using BFS
starting configuration

D

input tape

simulation tape

address tape
step 1:

- **input tape**: `a a b u u u u u u u u u u u ...
- **simulation tape**: `u u u u u u u u u u u u u u u u u ...
- **address tape**: `u u u u u u u u u u u u u u u u u ...`
order of branch simulation

\[ \Gamma_3 = \{1, 2, 3\} \]

order of traversal given by \((\Gamma_3)^*\) in short lex order

\[ \{\varepsilon, 1, 2, 3, 11, 12, 13, 21, 22, 23, 31, 32, 33, \ldots\} \]
simulating a branch

input tape

a a b ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ...  

simulation tape

a a b ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ...  

address tape

1 2 1 1 ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ⊔ ...
simulating a branch

1. simulate branch indicated by address tape
2. check if reject
   a. if so abort
3. check if accept
   a. if so accept
max number of children = 3
1. simulate branch indicated by address tape
2. check if invalid
   a. if so abort
3. check if reject
   a. if so abort
4. check if accept
   a. if so accept
Non-deterministic TM

Every NTM has an equivalent DTM
Enumerator

Turing Machine with Output

Sipser  
**Figure 3.20**  
Schematic of an enumerator
A language is Turing recognizable iff some enumerator enumerates it.

Given an enumerator \( E \) design a TM \( M \) that recognizes the language enumerated by \( E \).

Given a TM \( M \) design an enumerator \( E \) that enumerates the language recognized by \( M \).
Given an enumerator E design a TM M that recognizes the language enumerated by E

What does this mean?
A string \( w \) should be accepted by M iff it is (eventually) enumerated by E
Given an enumerator E design a TM M that recognizes the language enumerated by E

M =
“On input w
1. Run E. Every time E writes a string on its tape compare it with w.
2. If they match, accept. Else continue running E.”
list $\Sigma^*$ in short lex order
$S_1, S_2, \ldots$

Given a TM $M$ design an enumerator $E$ that enumerates the language recognized by $M$

$E =$
1. Run $M$ on strings $s_1, s_2, \ldots$
2. If $M$ accepts any $s_n$
   print $s_n$."

Try #1