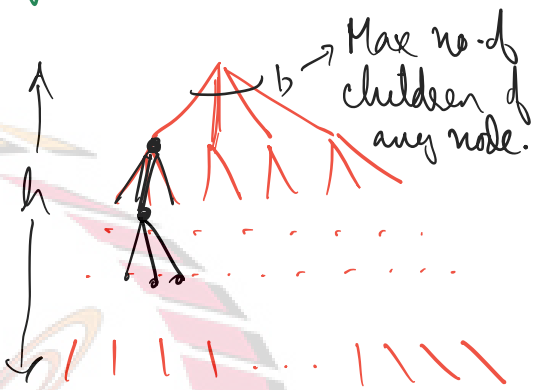
Equivalence of NTM and DTM

Theorem 3.16: Every NTM is equivalent to a DTM.

We need to search for an accepting computation path. How do we search in this tree?

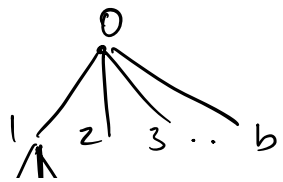


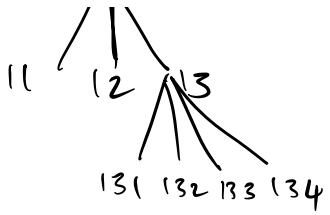
Computation Tree

DFS: We may go down an infinite path.

BFS: This works. We are sure to find if there is an accepting path. (only on the implicit graph. Config graphs are not explicitly constructed).

Check every node. Let b be the maximum no. of children possible for any configuration.



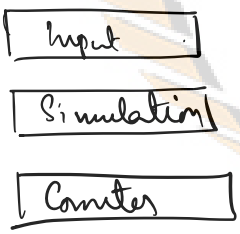


0. Root
1. Children of root 1, 2, ... b
2. 11, 12, ... 1b, 21, ... 2b, ... b1, b2, ... bb
3. 111, 112, ... 11b, 121, ... bbb.

∴ and so on.

Not all of these configurations may be valid.
There could be configurations with $< b$ children.

We will use a 3-tape DTM to simulate the NTM.



123112
123113
123121

- Initially tape 1 has input. Tapes 2 and 3 are empty.
- Copy tape 1 to tape 2.
- Simulate the NTM on tape 2. Use tape 3 as a counter to guide the choices. Accept if an accepting configuration is encountered.
- Replace tape 3 with the lexicographically next string. (Increment the count).

– Reject if all computations of the same length lead to reject.

if and only if

Cor 3.18: Turing recognizable $\iff$ Recognized by an NTM

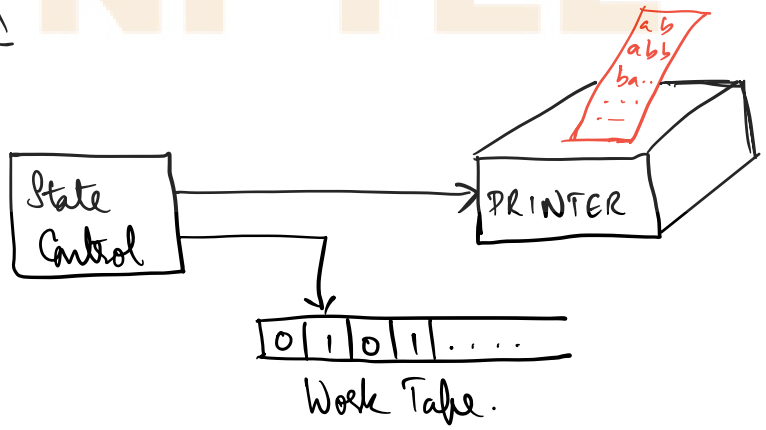
Cor 3.19: Decidable $\iff$ Decided by an NTM.

↓

All computation paths must halt.

No path must loop.

Enumerator



→ Starts with a blank tape.

→ Prints output one by one.

→ Language recognized by the enumerator E
= The set of strings printed out.

Theorem 3.21: A language is Turing recognizable if and only if some enumerator enumerates it.

Try Exercise 3.4: Give a formal definition of an enumerator. Consider it to be a type of two-tape Turing machine that uses second tape as a printer.

Equivalence of the TM models

We've seen different TM models. But all of them share the same computation power. Any model with unrestricted access to unlimited memory (along with other reasonable assumptions) are equivalent to TM's.