NONDETERMINISTIC

TURING MACHINES

Abhijit Das

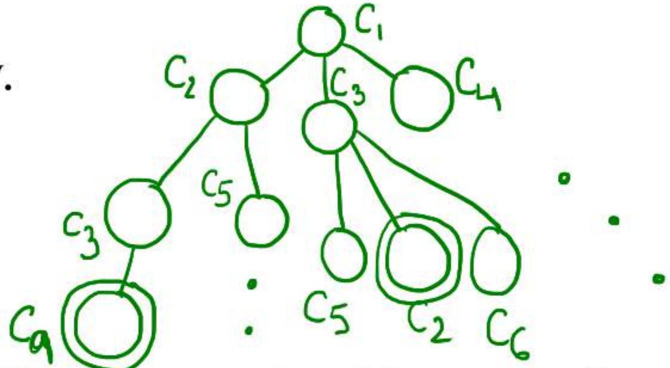
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur

March 26, 2020

Nondeterminism

- By definition, Turing machines are deterministic.
- TM = DTM = Deterministic Turing machine.
- NTM = Nondeterministic Turing machine.
- An NTM has $\delta(p,a) = \{(q_1,b_1,d_1),(q_2,b_2,d_2),\ldots,(q_k,b_k,d_k)\}$ for some $k \geqslant 0$.
- If $k = 0$, the machine gets stuck.
- If $k \geqslant 2$, the machine chooses one of the k moves nondeterministically.
- The maximum value of k over all combinations of p and a is called the (maximum) **fanout**. Denote the maximum fanout by f.
- The machine accepts if and only if **some** sequence of nondeterministic choices lets the machine reach the accept state.
- Other nondeterministic choices may lead to the stuck or a looping condition or even to the reject state.

Computation Histories



- Each sequence of choices gives a computation history.
- A history may terminate in the accept/reject state or in any other state in the stuck condition.
- Looping leads to an infinite history.
- The NTM **accepts** an input if and only if there is a finite computation history ending in the accept state.
- **Explicit reject:** All computation histories are finite, and none ends in the accept state.
- **Implicit reject:** No accepting history along with one or more infinite histories.
- A specific reject state is not needed for an NTM (even if total).
- The language of an NTM is the set of all input strings it accepts.
- An NTM is called **total** if all histories on all inputs are finite.
- All histories start with the initial configuration, and form a (potentially infinite) computation tree of configurations.

A Nondeterministic Compositeness Test

- A two-tape NTM for $\{a^n \mid n \text{ is composite}\}$.
- Input a^n is supplied to the first tape.
- **Initial check:** Try to copy two a's from the first tape to the second. If the attempt fails ($n = 0, 1$) or if no other symbol is left in the input ($n = 2$), reject and halt.
- If the machine is here, both the heads are pointing to the third cell (third a in the first tape and the first blank cell in the second tape).
- **Non-deterministic copying stage:** If there are more a's left in the input, make a non-deterministic choice.
  - Copy another a to the second tape, and move both the heads right.
  - Go to the division stage.
- **Division stage:** Suppose d number of a's are copied to the second tape.
  - Check whether the first head points to a blank cell ($d = n$). If so, reject and halt.
  - Otherwise, repeatedly subtract d from n. If some subtraction run erases the entire first tape, accept and halt. If some subtraction run prematurely ends, reject and halt.

Encoding Configurations

- The future work of a Turing machine (DTM/NTM) depends on:
  - State of the finite control.
  - Content of the tape.
  - The position of the head.
- A configuration is specified as the string $(p, u\underline{a}v)$, where $p \in Q$, $u, v \in \Gamma^*$, and $a \in \Gamma$. The underline represents the position of the head.
- We assume that $Q \cap \Gamma = \emptyset$.
- Then the configuration can be encoded as the string $upav \in (Q \cup \Gamma)^*$.
- The initial configuration on input w is $s \triangleright w$.
- An accepting configuration is $utav$ for any u, a, v.

Simulation of an NTM by a DTM

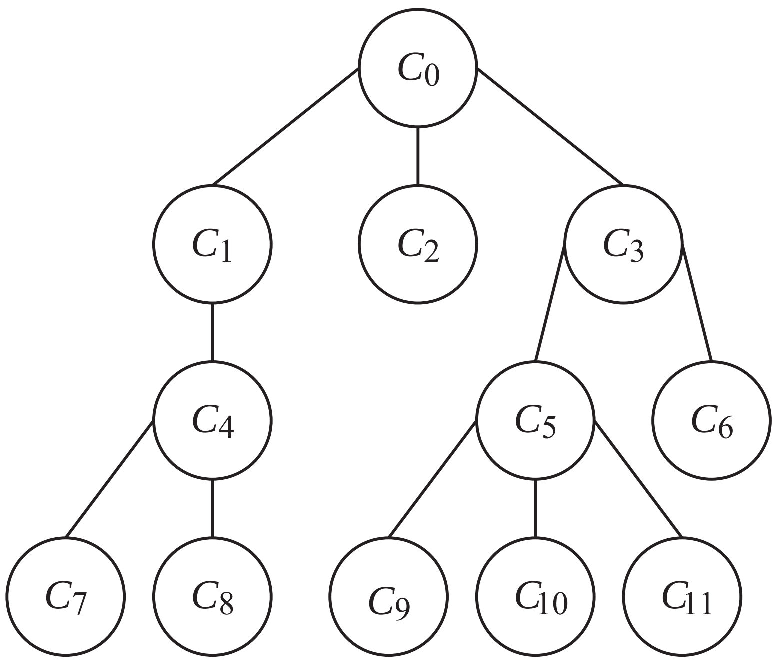
Theorem

For every NTM N, there exists a DTM D with $\mathscr{L}(D) = \mathscr{L}(N)$.

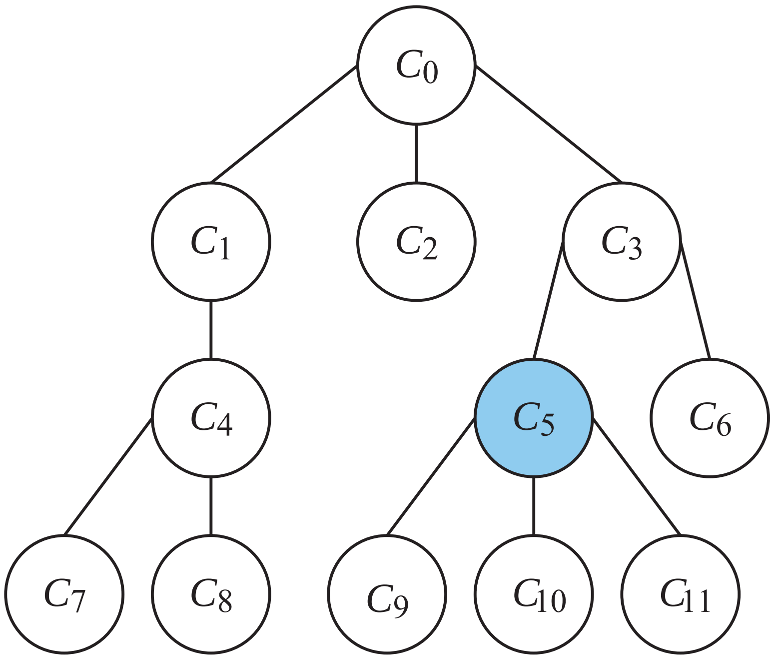
Theorem

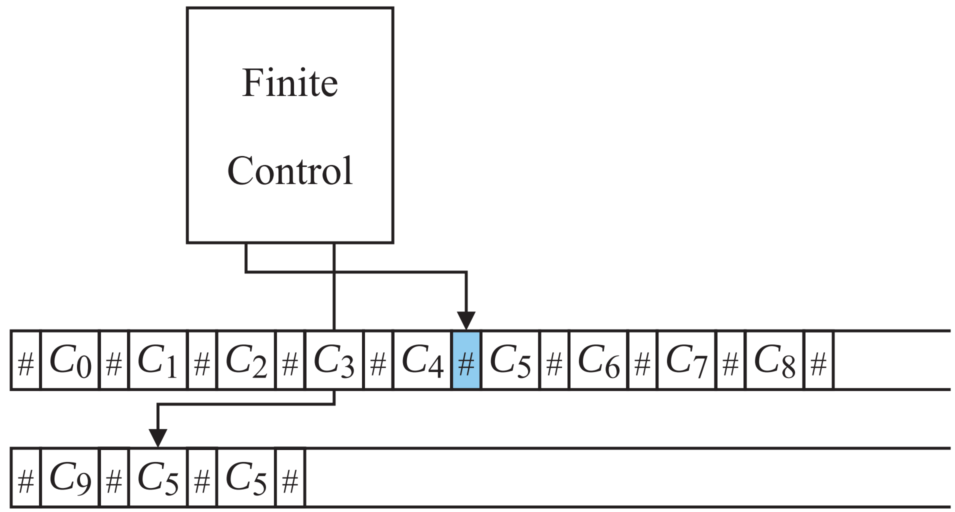
For every total NTM N, there exists a total DTM D with $\mathscr{L}(D) = \mathscr{L}(N)$.

Simulation of an NTM by a DTM



Simulation of an NTM by a DTM





Simulation of an NTM N by a DTM D

- D has the information of N in its finite control.
- D is a two-tape machine.
  - The first tape is used to implement a queue of configurations of N.
  - The second tape is used to compute next configurations of N.
  - The configurations are separated by a separator #.
  - One of the configurations is active denoted by a marker on the preceding #.
- The input w of N is given on the first tape of D.
- D uses the second tape to generate $C_0 = s \rhd w$.
- D replaces w by $\#C_0\#$ on the first tape.
- D then enters a loop.

Simulation of an NTM N by a DTM D

- D locates the current active configuration C_i.
- If there is no active configuration, D rejects and halts.
- Otherwise, D finds from C_i the state p of N and the tape symbol a scanned by the head of N.
- If $p = t$, D accepts and halts.
- D consults the transition table of N to identify $\delta_N(p,a) = \{(q_1,b_1,d_1),(q_2,b_2,d_2),\ldots,(q_k,b_k,d_k)\}$.
- D makes k copies of C_i to the second tape.
- D converts each copy by a transition possibility.
- D then copies the new configurations at the end of the first step.
- Finally, D moves the active marker from the current # to the next #.

More about the Simulation

- D makes a breadth-first traversal in the computation tree of N.
- If N is total, then D is total too.
- Let n be the number of steps in the longest computation history of N on some input.
- Let $f \geqslant 2$ be the maximum fanout of N.
- D needs to generate at most $1 + f + f^2 + \cdots + f^n = \frac{f^{n+1} - 1}{f - 1}$ configurations of N.
- The running time of N is taken as n.
- The running time of D is $\mathrm{O}(lf^n)$, where l is the longest configuration of N.
- It is not known whether the exponential slowdown of the simulation can be avoided in all cases.