W11L3 Turing machine runtimes and NP

Friday, April 17, 2020 9:19

Def: The class P is the set of all languages decidable in **polynomial** time (on a one-tape deterministic Turing machine).

Let's consider the extensions of Turing machines that we worked with previously.

Turing machines with L/R/S:
Idea: you can start with a LRS Turing machine and convert it to a L/R only Turing machine that does the same thing.
When we did this conversion, we basically extended every transition that did "stay" into 2 transitions "right and then left again".
If we add a runtime consideration to this proof, what happens?
If we had a LRS Turing machine which ran in worst-case runtime $O(t(n))$ on inputs of length n,
then this conversion will produce a LR only Turing machine which, in the worst case, takes 2 steps for every 1 step in the original Turing machine. So our new runtime will be $O(2 \cdot t(n))$. Notice that this is also $O(t(n))$.

Turing machines with k tapes:
Idea: you can start with a k −tape TM that runs in $O(t(n))$ steps and convert it to a 1-tape TM by storing the k tapes on the 1 tape, and then for each transition, we scanned the tape to find the position of the k tape heads and then scanned again to update all k tape heads.
If we add a runtime consideration…
For one transition of the k-tape TM, we need to scan the entire tape (all the square that are getting used) in order to find the information we need, and we need to scan again to update the information.
One scan of all the tape squares being used will take $O(t(n))$ time or $O(n)$ time, whichever is bigger.
Each transition of the k-tape TM becomes 2 scans of all the tape squares, so overall our 1-tape TM will have runtime $O(t(n) \cdot \max(t(n), n))$.

(theorem 7.8 on page 282 of Sipser)

So overall, if the k-tape TM was in polynomial time, then the 1-tape TM will also run in polynomial time.

Nondeterministic TMs:
Idea: start with a nondeterministic TM and convert it to a deterministic TM which checks all of the branches.

Theorem 7.11: Every $O(t(n))$-time one tape nondeterministic TM has a deterministic equivalent one-tape TM which runs in time $2^{O(t(n))}$.

Def: NP is the set of languages decidable in polynomial time by a one-tape nondeterministic Turing machine.

$NTIME(t(n)) =$ the set of languages decidable in time $O(t(n))$ by a one-tape nondeterministic Turing machine

$$NP = \bigcup_{\substack{f(n) \\ \text{polynomial}}} NTIME(f(n))$$

$$coNP = \{ L \mid \overline{L} \in NP \}$$

If $L \in P$ then there is a polytime deterministic decider D for L.

Using D we can build a polytime deterministic decider for $\overline{L}$.

$\overline{L} \in P$ so $\overline{L} \in NP$ so $L \in coNP$.