Chapter 16-NP

Monday, January 9, 2023
2:57 PM

EXP is the group of problems that require (at least) exponential time to solve.
NP is the group of problems where the best known algorithm is exponential, but no one has proved that it is impossible to construct a polynomial-time algorithm.

Sometimes it's tough for a program to find parse trees.
If a sentence has *n* words, it takes **O(n^3)** time for a properly-designed parsing algorithm to find a parse tree for the sentence or determine that none exists.
However, generating **all** parse trees for such a sentence takes **exponential** time.

This is not because computers are dumb, but because there are simply too many possibilities. (so parse trees calculation is EXP, not NP)

Consider sentences that end in a large number of prepositional phrases, like:
Prof. Manfroi killed a zombie with a red hat near the Academic Office on the ground floor by the statue with a large sword.

Each prepositional phrase can belong to either the main verb or some other noun phrase in the sentence. (e.g. who has the large sword? Manfroi or the statue?)

Another example of NP is ***Graph Colorability***:
Known algorithms like the *greedy coloring algorithm* in chapter 11 work well for most cases. However, for some graphs, all known algorithms take exponential time. We just don't know if there is some clever trick that can make the process faster.

A computational problem is in the set ***NP*** when an algorithm can provide **justifications** of "yes" answers, where each justification can be checked in **polynomial time**.
(This is why parse trees are EXP, because the raw answer without justification has exponential length)

Co-NP is the set of problems for which we can give polynomial time checkable justifications for **Negative Answers**.

P is the set of problems for which we can solve with polynomial time.

(it is unclear whether the sets NP, Co-NP, and P are the same set)

A ***Boolean Circuit*** is a type of graph consisting of a set of ***Gates*** connected together by wires. (*ECE Moment*)
Algorithms that are used to find bad circuit design (ones that yield bad output patterns) are called circuit satisfiability algorithms.
These algorithms are generally exponential and in NP (2^n possible input patterns and easily-verifiable justification for solutions)

However, circuit safety, or proving a circuit is safe from bad outputs, is in co-NP. (since proving a negative answer *seems to* require walking through all exponentially many input patterns and showing why none of them produces the bad output pattern)

NP Complete:
It can be proven that finding a polynomial-time algorithm for either circuit satisfiability or graph colorability would imply that all problems in NP have polynomial-time algorithms. Such problems are called ***NP-Complete.***