Review of Decision Problems

What is a Decision Problem?

- **A decision problem is a defined by a property S about some discrete universe of discourse, e.g., Natural Numbers, Graphs, Programs. We will call this universe U.**
- **Given S, our goal is to determine which elements of U have property S and which do not.**
- **A property S is decidable (recursive, solvable) if there is a algorithmic predicate χ_S, such that for any element x of U, $\chi_S(x)$ is true if x has property S and false otherwise. χ_S is called the characteristic function of S**

What are P and NP?

- **Abstractly, if a predicate χ_S that decides property S can be written to run in polynomial time on a single processor, then S is in P.**
- **Concretely, if a predicate χ_S that decides property S can be written to run in polynomial time on a deterministic Turing machine, then S is in P.**
- **Abstractly, if a predicate χ_S that decides property S can be written to run in polynomial time using an arbitrary number of processors, then S is in NP.**
- **Concretely, if a predicate χ_S that decides property S can be written to run in polynomial time on a non-deterministic Turing machine, then S is in NP.**
- **It is typically more useful to employ the following for NP**
  if a predicate χ_S can be written to check a proposed solution to any instance of P in polynomial time on a deterministic Turing machine, then S is in NP.

What is NP-Complete (NPC) and Why is SAT NPC?

- S is NP-Complete if all NP problems can be reduced to S is polynomial time.
- Given a non-deterministic Turing Machine, M_S, that decides some S in NP and an instance, x, of S's universe, we can show that there is a tableau that describes all traces of computations in M_S, and that each trace can easily be inspected to see if it represents an acceptance or rejection of S.
- The traces described above are all of polynomial length since they are describing a polynomial time machine computation.
- Given M_S, a polynomial time deterministic algorithm can be presented that produces an instance of SAT which is satisfiable iff there is an accepting trace of M_S's computation when it is presented the instance x.
- Thus, any arbitrary instance of NP, S, is polynomial time reducible to SAT.

What are some other problems in NP-C?

- **SAT is reducible in polynomial time to 3-SAT, so 3-SAT is in NP-Hard.**
- **As 3-SAT is in NP and is NP-Hard, it is in NP-C.**
- **3-SAT is polynomial time reducible to SubsetSum, so SubsetSum is in NP-Hard**
- **As SubsetSum is in NP and is NP-Hard, it is in NP-C.**
- **SubsetSum is polynomial time reducible to Partition, so Partition is in NP-Hard**
- **As Partition is in NP and is NP-Hard, it is in NP-C.**
- **More to come …**