NP complete:

1) in NP

2) every problem in NP reducible to it.

Transitivity of p-time reduction implies

NP complete iff

1) in NP

2') some NP-complete problem reducible to it.

We need one NP-complete problem to get started.
$\text{Sat} \in \text{NP}$: proof = satisfying assignment
$
\text{Graph coloring} \in \text{NP}$: proof = coloring
$k$-clique, $k$-vertex cover, $k$-independent set $\in \text{NP}$

Tautology $\in \text{co-NP}$ ("no" instances have a proof)
Cook-Levin Theorem: Sat is NP-complete.

Given any p-time verifier, construct (in p-time) an instance of Sat s.t. verifier answers "yes" iff formula is satisfiable.

Verifier: Turing Machine

In one step, machine can write a symbol, move head one position, change state.

What to do is based on state, symbol read.

Fixed # of states, fixed # of tape symbols, including blank; start state, "yes" state, ("no" state)

Explicitly given polynomial time bound p(n).
Input (of size $n$) is a "yes" instance iff for some "proof" and given input, the machine reaches "yes" state within $p(n)$ steps from start state.

Must construct a formula that is satisfiable iff this happens.

Note: input is specified, proof is not (non-deterministic part)

Proof can't exceed length $p(n)$; machine can't get farther in $p(n)$ steps.

Can assume machine loops in "yes" state; if ever in "yes", will be in "yes" at step $p(n)$.
States: 1, ..., y  \quad 1 = \text{start}, \ y = \text{yes}

Symbols: 1, ..., z  \quad 1 = \text{blank}

Tape cells, \ -p(n), ..., 0, ..., p(n)

Time: 0, 1, ..., p(n)

Variables for formula:

\begin{align*}
\text{hit}_i & : \text{true if head on tape cell } i \text{ at time } t \\
    & -p(n) \leq i \leq p(n), \ 0 \leq t \leq p(n) \\
\text{st}_j & : \text{true if state } j \text{ at time } t \\
    & 1 \leq j \leq y, \ 0 \leq t \leq p(n) \\
\text{c}_i & : \text{true if tape cell } i \text{ holds symbol } k \text{ at time } t \\
    & -p(n) \leq i \leq p(n), \ 1 \leq k \leq z, \ 0 \leq t \leq p(n)
\end{align*}
What does the formula need to say?

At most one state, head position, and symbol per cell at each time:

\[(\overline{h_{ik}} \lor \overline{h_{ij}}) \quad i \neq i', \quad \text{all } t\]
\[(\overline{c_{ij}} \lor \overline{c_{jj}}) \quad j \neq j', \quad \text{all } t\]
\[(\overline{c_{ik}} \lor \overline{c_{ik'}}) \quad k \neq k', \quad \text{all } i, \quad \text{all } t\]

Correct initial state, head position, and tape contents:

\[1_{00} \lor 0_{01} \lor 0_{10} \lor 1_{10} \lor \cdots \lor c_{1k_0} \lor c_{2k_0} \lor \cdots \lor c_{nk_0}\]
\[\cdots \lor c_{(n-1)10} \lor \cdots \lor c_{(n+1)10}\]

Input is \(k_1, k_2, \ldots, k_n\), rest of right side of tape is blank.
Correct final state: $s_p(n)$

Correct transitions:

E.g. if machine in state $j$ reads $k$, it then writes $k'$, moves head right, and changes to state $j'$:

$$s_{jt} \land h_{it} \land c_{ikt} \Rightarrow s_{j't+1} \land h_{i't+1} \land c_{i'k't+1}$$

($\Rightarrow$ = "implies") (for each $i, t$)

$$h_{it} \land c_{ikt} \Rightarrow c_{i'kt+1}$$ (for $i \neq i'$, each $k, t$)

(unread tape cells are unaffected)

CNF?
\((x \land y \land z) \Rightarrow (a \land b \land c)\)

\[\Rightarrow\]

\(((x \land y \land z) \Rightarrow a)\]

\(((y \land z) \Rightarrow a)\]

\(((z) \Rightarrow a)\]

\[\Rightarrow\]

\((\bar{x} \lor \bar{y} \lor z \lor a)\]

\((\bar{x} \lor \bar{y} \lor \bar{z} \lor b)\]

\((\bar{x} \lor \bar{y} \lor \bar{z} \lor c)\]
Any proof that gives a "yes" execution
gives a satisfying assignment, and
vice-versa.

Conclusion: SAT is NP-complete
(and k-coloring, k-clique, k-independent set,
k-vertex cover)
Subset Sum is NP-complete

Given n integers, and a target \( k \), is there a subset that sums to exactly \( k \)?

\[
\{ 2, 5, 6, 8, 9, 12 \} \quad k = 31
\]

yes: 5, 6, 8, 12

( no for \( k = 30 \) )

In NP: subset is proof (verifiable in p-time)

Some NPC problem reducible to subset sum

reduce 3-CNF sat to subset sum
Write numbers base 5

\((x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor y \lor z) \land (y \lor \bar{z})\)

\[\begin{array}{ccc}
C_1 & C_2 & C_3 \\
\hline 
 x & 1 & 0 & 0 & 1 & 0 & 0 \\
\bar{x} & 1 & 0 & 0 & 0 & 1 & 0 \\
y & 0 & 1 & 0 & 0 & 1 & 0 \\
\bar{y} & 0 & 1 & 0 & 1 & 0 & 0 \\
z & 0 & 0 & 1 & 0 & 1 & 0 \\
\bar{z} & 0 & 0 & 1 & 1 & 0 & 0 \\
\hline
1 & 1 & 1 & 4 & 4 & 4 & 4 = b \text{ Required sum}
\end{array}\]

Dummies to get clause columns to sum to 4

Interpret each row as a base-5 (or base-10) number.

Subset sum has a solution iff formula is satisfiable.