Wrap-up of NP, coNP, NP-completeness

Chromatic number of a graph $G$ is minimal number $k$ of colors $s.t. G$ is $k$-colorable.

$$\text{Chromatic Num} = \exists k \geq 1 \text{ s.t. } k \text{ is } G \text{'s chromatic number}$$

$$\text{Subsets } \mathcal{P}_n = \{ \langle s, t \rangle \mid s = \{ s_1, \ldots, s_k \text{ is a set of integers written in decimal}, t \text{ is an int. set, } \exists T \subseteq s \text{ s.t. } \}$$

$$\mathcal{P}_n = \sum s \cdot t$$

$$\text{Hamiltonian Path } = \{ \langle G, s, t \rangle \mid \text{G is a directed graph s.t. } \exists \text{ a path from } s \text{ to } t \text{ that visits every vertex of } G \text{ exactly once} \}$$

$$\text{NP-SAT } = \{ \langle C, n \rangle \mid \text{C is a set of triples of literals over } n \text{ Boolean variables } x_1, \ldots, x_n, \text{ and } \exists \text{ an assignment } a \text{ on } x_1, \ldots, x_n \text{ s.t. every literal has a true and a false literal} \}$$

Chromatic Num not in NP unless something surprising happens.

NP = coNP.

Claim 1: CN (Chromatic Num) is NP-hard.

Proof: Let the CN we saw that 3SAT $\leq_p$ 3-colorability.

Reduction: on input $\phi,$ output $G$.

$$+ \text{ additional vertices edges.}$$

$G$ needs at least 3 colors, since there's a triangle. Therefore, if $\phi \in \text{3SAT, chromatic # of } G \geq 3$.
If $\phi \in 3SAT$, then $CN(G)$ is 4.

$3SAT \leq CN$ using the same reduction to compute $G$,

we output $(G, 3)$.

Thus, $CN$ is NP-hard.

Claim 2: $CN$ is coNP-hard.

**Proof:** Reduce from $(3SAT)^c$.

On input $\phi$, compute $G$ as in last class's reduction,

output $(G, 4)$.

*Analysis:* polytime, since it uses a previous reduction.

*Correctness:* $G \in 3SAT \Rightarrow CN(G) = 4 \Rightarrow (G, 4) \in CN$.

$\phi \notin 3SAT \Rightarrow CN(G) = 3 \Rightarrow (G, 3) \notin CN$.

Like in HW, $CN \not\in NP \Rightarrow NP = coNP$.

Claim: If $NP = NP$, then ChromaticNumber $\in P$.

**Proof:** If $P=NP$, we have a polytime algorithm $A_k$ deciding $k$-color.

On input $(G, k)$,

for $i = 0$ to $n = |V(G)|$,

run $A_k$ on input $(G, i)$.

If $A_k$ accepts and $i = k$,

accept, else reject.

3

Clique Contest, $CN$ with $4$ colors $T, F, S, N$,

we can color $G$.
If $NP = coNP$

![Venn diagram showing $NP = coNP$ and $coNP$-hard]

To show $CN \in NP \implies NP = coNP$,

1. $NP = coNP$
2. $coNP \leq NP$

E.g., to do 2, take $L \in coNP$ and show $L \in NP$,

by giving a polytime TM for $L$, as follows

On input $w$:

1. Reduce to a $P$-hard language, like $CN$, set $w \leq G(w)$
2. Use TM deciding $CN$ in poly time on $\langle G, w \rangle$

This gives an TM deciding $L$.

That $SubsetSum$ is $NP$-complete.

Note decimal is okay, we can reduce binary to decimal and vice-versa.

Unary $SubsetSum$ is in $P$.

pf: 1. $SS \in NP$. [Exercise!]
2. $3SAT \leq_P SS$.

Assume $n$ variables, $2^n$ literals in clauses $C_1 \ldots C_m$

<table>
<thead>
<tr>
<th>$S_2 = \overline{x}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_3 = \overline{\overline{x}}$</td>
</tr>
<tr>
<td>$S_4 = \overline{\overline{x}}$</td>
</tr>
<tr>
<td>$S_5 = \overline{x}$</td>
</tr>
<tr>
<td>$S_6 = \overline{x}$</td>
</tr>
</tbody>
</table>

Integers in $S$ correspond to literals, and clauses correspond to variables.

If $C_1 = x_1 \lor \overline{x}_2 \lor x_3$, then integer corresponds to $x_1, \overline{x}_2, x_3$.
For \( l \), let last \( m \) digits be \( 1 \), so each clause is satisfied.

Every variable must be either true or false so we want the first \( n \) digits to be 1.

I don't have integers that subtract by 2 from a given digit corresponding to a clause. Then just subtract one to get to \( t \), which has a 1 in each digit.

Book's idea: \( t \) has a 1 in the first \( n \) digits and 3 in the last \( m \).

Then we add integers that allow us to get to 3.

So we can just add "slack" elements of \( S \) to get each clause digit to 3.

On input \( \phi \) with \( n \) variables, \( m \) clauses, output a set \( S \) of \( 2n+3m \) integers of \( n+2m \) decimal digits.

Integer \( s_{i1} \) corresponds to literal \( x_i \): it has a 1 in position \( i \) and in positions \( n+j \) if \( x_i \in C_j \).

Integer \( s_{2i+1} \) is literal \( \overline{x}_i \): \( 1 \) in position \( i \) and in positions \( n+j \) if \( \overline{x}_i \in C_j \).

Integer \( s_{2i+2} \) has a 1 in position \( n+j \) and 1 in position \( n+i+j \) for \( 0 \leq a \leq 2 \).

\( t \) has a 1 in first \( n \) positions, 3 in next \( n \) positions, and 1 in next \( m \) positions.
Analyze polynomial # of digits, so this is poly time.

\[ \Phi \in 3SAT \implies \text{there's a satisfying assignment} \alpha_1, \ldots, \alpha_n \]

Choose elements of \( S \), \( s_i \) corresponds to \( x_i = \alpha_i \)

and the "slack" integers padding the classic digits to 3.

\[ <S, t> \in 3S \]

\[ <S, t> \in 3S \implies \text{exactly one slack integer is picked} \]

\[ \implies \text{each clause is satisfied} \]

Also, no conflicting literals are picked.

\[ \implies \text{there's a satisfying assignment} \]

\[ \implies \Phi \in 3SAT \]

**Theorem:** Hamiltonian Path is NP-complete.

Variations on HP that are also NP-complete:

- Hamiltonian Cycle
- Undirected version of HP, HC

\[ \text{ide} \]"u"

\[ \varepsilon \]

\[ \text{Vin } \text{ wait } \text{ wait } \text{ wait } \text{ Vin} \]

**Fact:**

1. HP \( \in \) NP
2. NP-hard reduction from 3SAT.

Say \( \Phi \) has \( n \) variables, \( m \) clauses.

\[ \begin{array}{c}
X_1 \leftarrow r_1 \\
X_2 \leftarrow r_2 \\
\vdots \\
X_n \leftarrow r_n \\
\end{array} \]

Can only go to \( C_i \) in one direction.