21 Apr 2025 – Theory of Computation – Week 15

$$P := \mathrm{DTIME}(n^k)$$

$$NP := \mathrm{NTIME}(n^k)$$

Thm: $\mathrm{DTIME}(t(n)) \subseteq \mathrm{NTIME}(t(n)) \subseteq \mathrm{DSPACE}(t(n))$

where $t : \mathbb{N} \to \mathbb{N}$

$$EXP := \bigcup_{k>0} \mathrm{DTIME}\left(2^{n^k}\right)$$

Thm: $P \subseteq EXP$ $P \subseteq NP \subseteq EXP$

$NP \underset{?}{\subseteq} EXP$ ✓

$NTIME(n^k)$

in any one branch # of non-deterministic choices $\leq n^k$

Total no. of branches $\longrightarrow 2^{n^k}$ (assuming $\{0,1\}$)

Time taken: $2^{n^k} \cdot \underbrace{n^k}_{\text{simulation in particular branch}}$

$\text{EXP} \overset{?}{\subseteq} \text{NP}$

$2^{2^{n^k}}$

Q: Can allowing more resources allow you to solve more problems?

recognize more languages

YES

Time Hierarchy

If $\lim_{n \to \infty} \frac{T_1(n) \log T_1(n)}{T_2(n)} = 0$

then $\text{DTIME}(T_2(n)) \supsetneq \text{DTIME}(T_1(n))$

Space Hierarchy

If $\lim_{n \to \infty} \frac{S_1(n)}{S_2(n)} = 0$

then $\text{DSPACE}(S_2(n)) \supsetneq \text{DSPACE}(S_1(n))$

There is a gap b/w

$n \log n \xleftrightarrow[\text{nothing}]{} n$

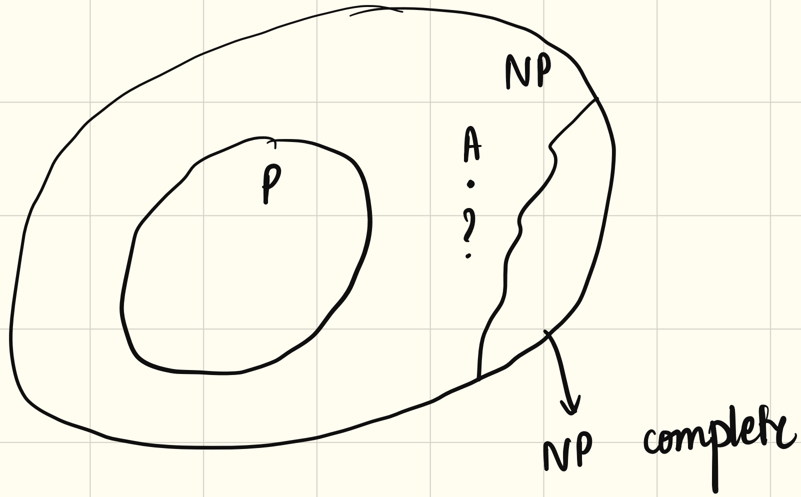
$n \log(\log n) \xleftrightarrow{\text{equivalent}} n$

From time hierarchy $\longrightarrow$ $P \subset EXP$

$\therefore$ One of these is strict

$P \subseteq NP \subseteq EXP$

Is $P = NP$? $\longrightarrow$ P vs NP problem



If $P \neq NP$, We need a language $A \in NP$ but $\notin P$

Polynomial-time many-one reduction

$f : \Sigma^* \longrightarrow \Delta^*$ is a poly-time computable function if $\exists$ a det. TM running in polynomial time and computing f.

a language $A \leq_p B$ if $\exists$ a poly-time computable function f s.t.

$$x \in A \iff f(x) \in B$$

<u>Thm</u>: if $A \leq_p B$ and $B \in P$

then $A \in P$.

<u>Thm</u>: If $A \leq_p B$ and $B \in NP$

then $A \in NP$

Suppose $\forall L \in NP, \quad L \leq_p A$ — A is the hardest problem in NP

further $A \in P$.

If you can solve A, you can solve everything in NP.

then $NP \subseteq P$

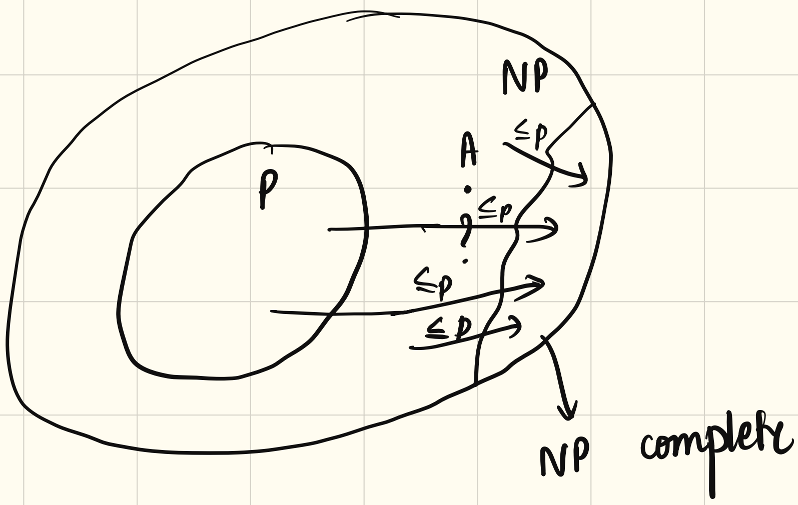
Def^n: A language A is said to be <u>NP - hard</u> if $\forall L \in NP, \quad L \leq_p A$.

Def^n: A language A is said to be NP - complete if

1) $A \in NP$

2) A is NP - hard

EXP can also be NP - hard



Suppose a language is NP-hard and $A \le_p B$ then B is NP-hard

If $A \le_p B \le_p C$

$\Rightarrow A \le_p C$

22 Apr 2025

P - hard ?

A is $\mathcal{C}$ - hard

if $\forall L \in \mathcal{C}$, $L \leq_p A$

reduction itself allows Polynomial time

B/w variants of TM

P and NP remain same

Thm: Let $L \in DTIME_{2\text{-tape}}(t(n))$

then $L \in DTIME_{1\text{-tape}}(t(n)^2)$

k-tapes $\rightsquigarrow$?

??

DTIME (n) $\xrightarrow{\text{maybe}}$ DTIME(n^2)

Two tape

Boolean variables

$$x_i \in \{\overset{\text{false}}{0}, \overset{\text{true}}{1}\} \;/\; \{T, F\}$$

$$x_i \wedge x_j \;,\quad x_i \vee x_j \;,\quad \neg x_i \,/\, \sim x_i \,/\, \overline{x_i}$$

Boolean formula

We say E is a boolean formula, over $\{x_1, \dots, x_n\}$ if

i) $E = x_i$, $i \in [n]$ or

ii) $E = E_1 \vee E_2 \,/\, E_1 \wedge E_2 \,/\, \sim E_1$

where E_1, E_2 are boolean formula

$$\psi(x_1, x_2, \dots, x_6) = \left((x_1 \vee x_2) \wedge \overline{x_3} \right) \vee \left(\overline{x_4} \wedge (x_5 \vee x_6) \right)$$

Q: Given a boolean formula $\psi(x_1, \dots, x_6)$. $\widetilde{\text{Decide}}$ if $\exists$ a $y \in \{0, 1\}^n$ s.t $\psi(y) = 1$

" $\psi(x)$ is satisfiable "

$$SAT := \{ \langle \psi \rangle \mid \psi \text{ is satisfiable} \}$$

$SAT \in NP$?

Non-deterministically choose a value to set for $\{x_1, \dots, x_n\}$

NP defn

① $\exists$ a NDTM running in poly-time

verifier machine

② $L \in NP$ if $\exists$ a DTM M running in poly-time s.t.

$\forall x \in \{0,1\}^*$

$x \in L \Leftrightarrow \exists z \in \{0,1\}^{p(|x|)}$

s.t. $M(x, z)$ accepts

$z \equiv$ proof

If $x \in L$, you can give a short proof

For SAT,

$$|z| = |x|$$

$$M\left(\langle \psi \rangle ,\ \underset{z\,=\,b}{\underline{x_1 = \psi_1 ,\ x_2 = \psi_2 ,\ \ldots}} \right)$$

$\Rightarrow$ if b satisfies $\langle \psi \rangle$ then M accepts
else reject.

Def^n k-clique $\equiv$ complete graph on k vertices.

Q. Given a graph $G = (V, E)$ and an integer k.
Decide if $\exists$ a k-clique in G?

$\binom{n}{k}$ choices

(not polynomial)

Given a sol^n

$\binom{k}{2}$ checks

23 Apr 2025

SAT := $\{ \langle \varphi \rangle \mid \varphi$ is a satisfiable boolean formula $\}$

Cook – Levin Thm: SAT is NP-complete.

$\in$ NP
relatively
easy
to prove

is NP-hard

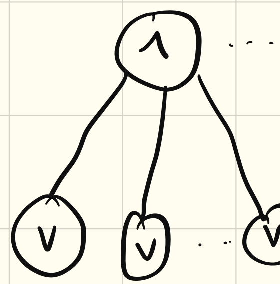
also NP - complete

CNF - SAT := $\{ \langle \psi \rangle \mid \psi$ is a satisfiable CNF formula $\}$

CNF := conjunctive normal form

literals

$$\underbrace{(x_1 \vee x_2 \vee \overline{x_3})}_{\text{clauses}} \wedge (x_5 \vee x_6) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_5})$$



NP complete

$\underbrace{3\ \text{SAT}}$:= { ⟨ψ⟩ | ψ is a satisfiable 3-CNF formula }

k-CNF := CNF where every clause contains $\leq k$ literals

<u>k-clique</u> := { ⟨G, k⟩ | G is undirected simple graph that has a clique of size k }

Thm: k-clique is NP-complete $\longrightarrow$ $\in$ NP ✓ previous class

Proof: It suffices to reduce a known NP-hard problem in poly-time to k-clique.

$$3\ SAT \xrightarrow{\quad f \quad} k\text{-clique}$$

$$x \in 3\ SAT \iff f(x) \in k\text{-clique}$$

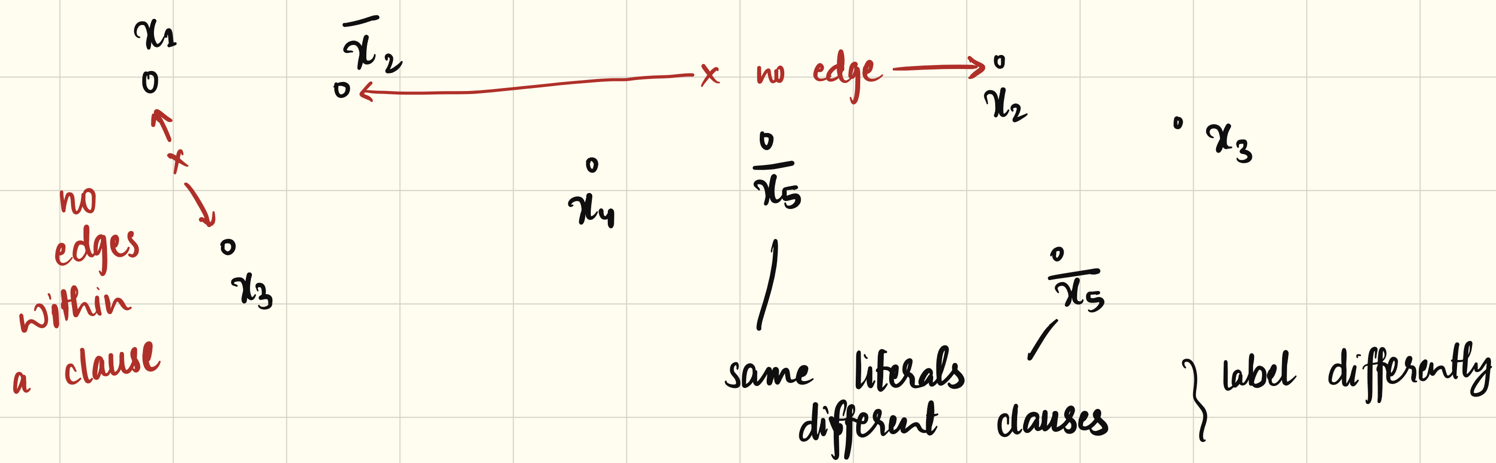
$$\psi \in 3\text{-CNF} \xrightarrow{f} f(\psi) = (G, k)$$

ψ is satisfiable $\implies$ G has a clique of size k

ψ is unsatisfiable $\implies$ G does not have a k-clique.

$$(x_1 \vee \overline{x_2} \vee x_3) \wedge (x_4 \vee \overline{x_5}) \wedge (x_2 \wedge x_3 \wedge \overline{x_5})$$

G_1



Total no. of vertices in G_1 = no. of literals
(w/ multiplicity)

- no edges b/w vertices of the same clause.
- no edge b/w x_i and $\overline{x_i}$
- all other edges are present.

Claim 1: φ satisfiable $\Rightarrow$ G has a clique of size k.

$\downarrow$

all clauses evaluate to 1

$\downarrow$

at least one literal = 1

choose any one literal from each clause $\nearrow$ form a clique of size $k := \#$ of clauses

<u>Claim 2</u>: G has a clique of size k $\Longrightarrow$ ψ is satisfiable.

A clique of size k exists $\Longrightarrow$ every vertex belongs to different clauses

$\oplus$

no contradictory vertices

$\swarrow$

ψ satisfiable

$f \longrightarrow$ polynomial

$3\,SAT \leq_p k\text{-clique}$

works with

arbitray k.

Thm :

$$k\text{-clique} \leq_p \underbrace{k'\text{-Independent set}}$$

given a graph does it contain atleast k vertices w/ no edges b/w them

$$G, k \xrightarrow{f} \bar{G}, k$$

complementary graph

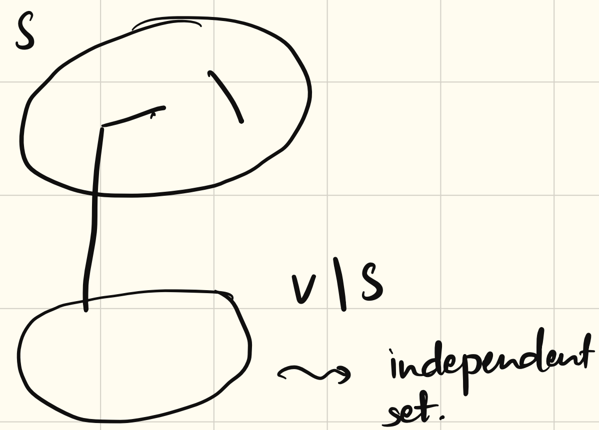
Vertex cover of a graph G is a subset $S \subseteq V$ s.t $\forall\ e = (u, v) \in E$, one of u or $v \in S$.

$I.S \leq_p V.C$ → is there a vertex cover of size $\leq k$

$G, k \xrightarrow{f} G', k'$

independent set of size at least k $\iff$ G', k' has a vc of size $\leq k'$

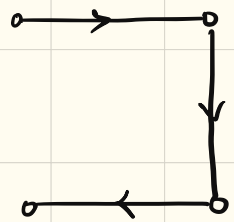
$G' = G$, $k' = n - k$



Thm: DHAMPATH is NP-complete

DHAMPATH $:= \{ \langle G \rangle \mid G$ is a directed graph that contains a hamiltonian path $\}$

A hamiltonian path is a simple path that visits every vertex.



Thm: 3SAT $\leq_p$ DHAMPATH

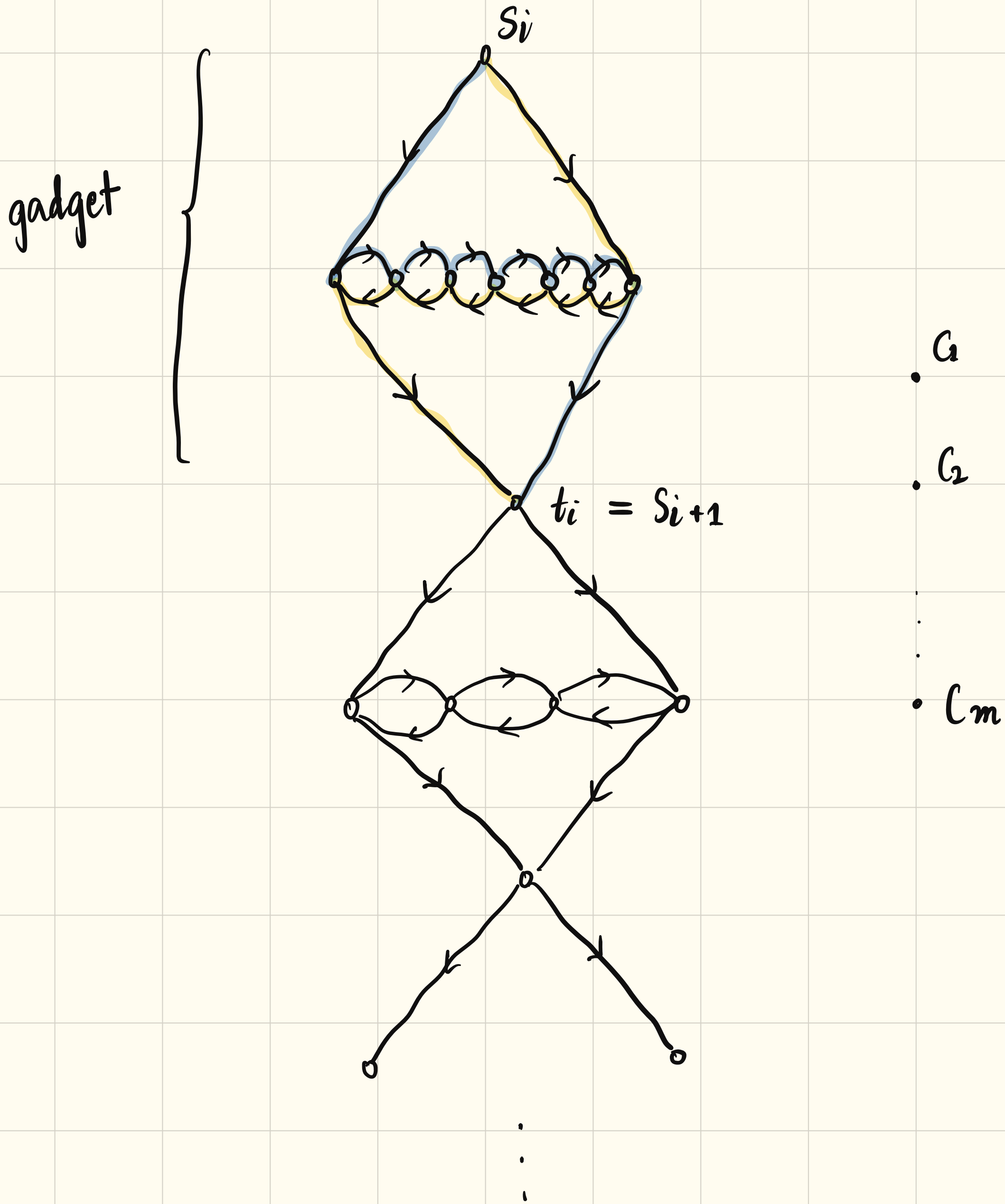
a 3-CNF ψ $\xrightarrow{f}$ a directed graph G

$\psi \in$ 3SAT $\iff$ $G \in$ DHAMPATH

<u>Proof</u>: $\psi = (x_1 \vee \overline{x_2} \vee x_3) \wedge \ldots$

for every variable $x_i \in \psi$

construct the following graph



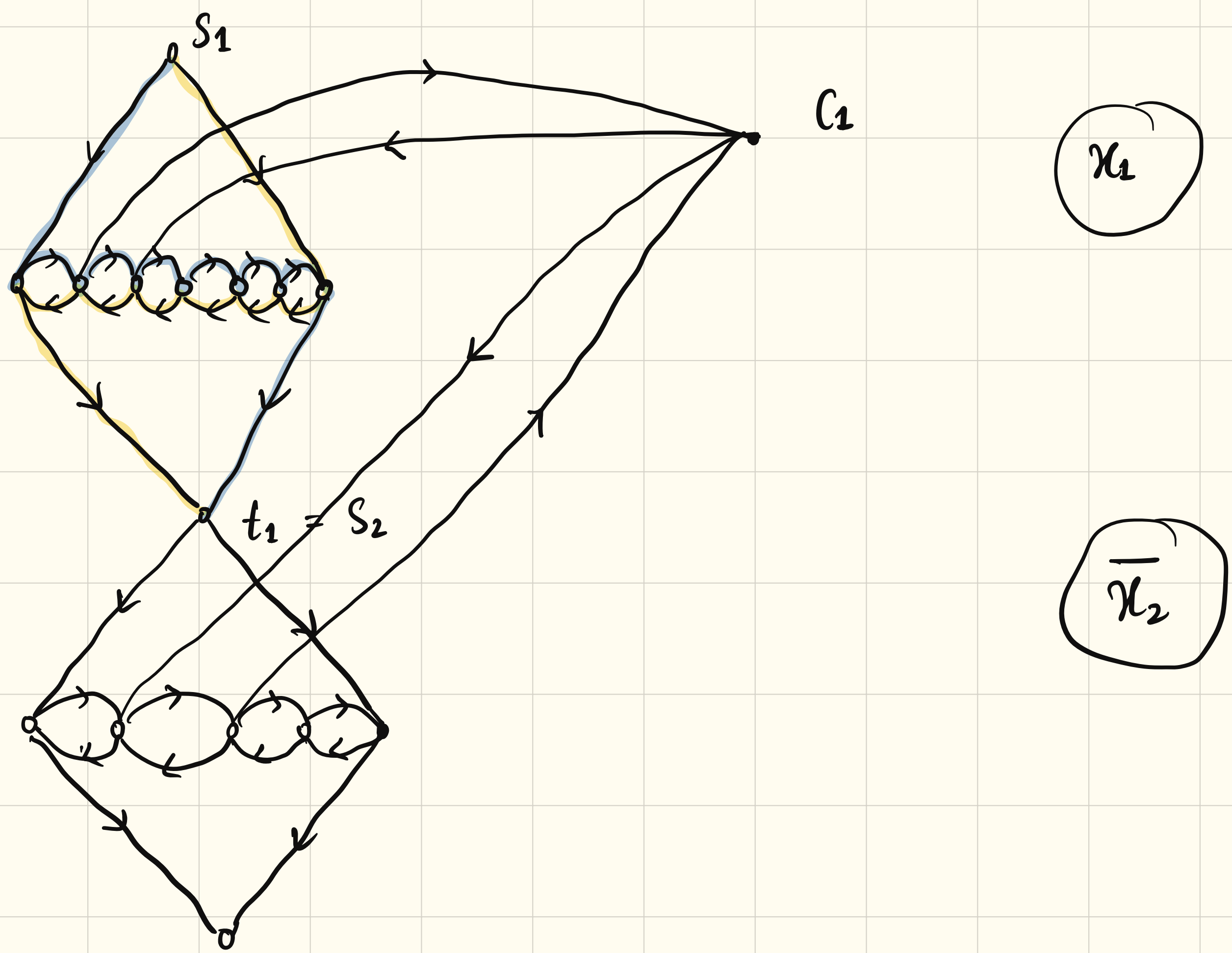
2 Hamiltonian paths

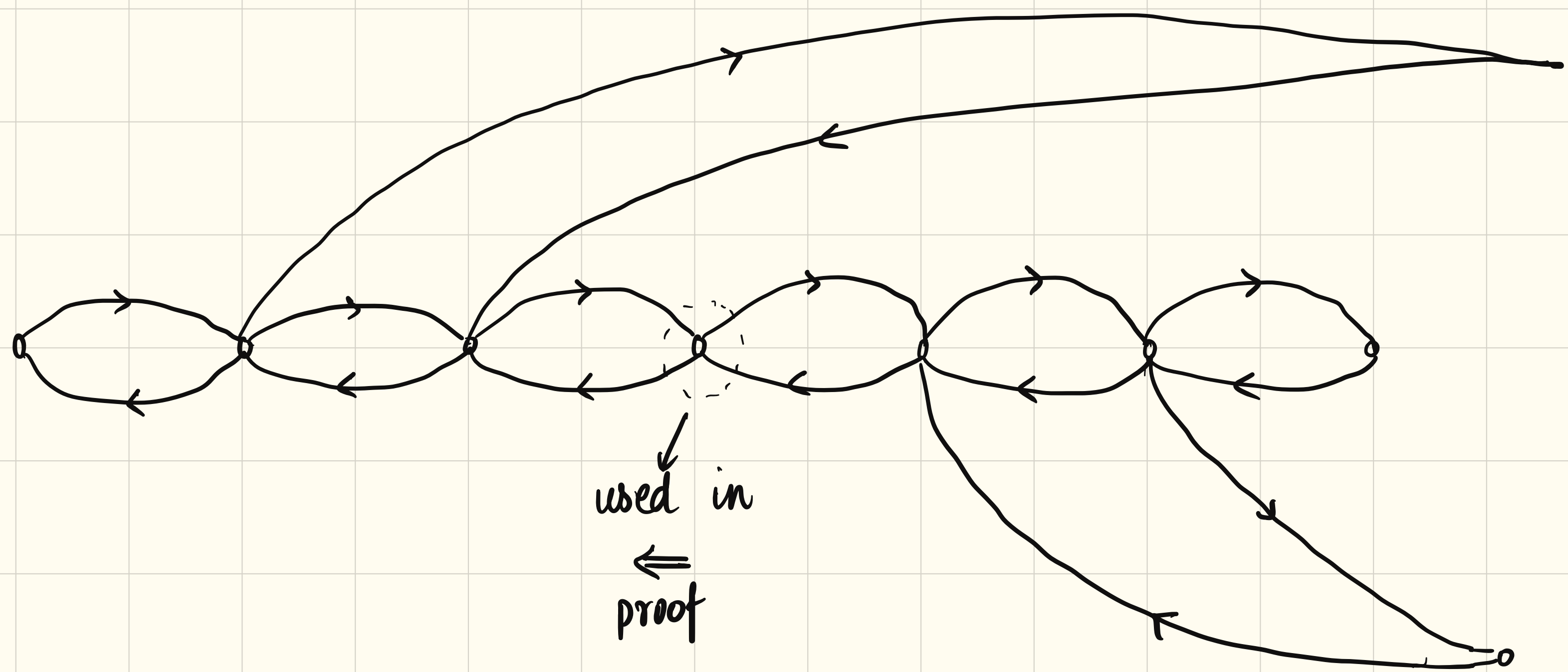
"L-R Ham path" = "$x_i = 1$"

"R-L Ham path" = "$x_i = 0$"

No. of hamiltonian paths possible = 2^n

correspond to 2^n different settings for ψ.


s_1
c_1
$t_1 = s_2$
$\mathcal{H}_1$
$\overline{\mathcal{H}_2}$



A variable can occur at most m times in ψ

$\therefore$ $3m + 2$ vertices in chain

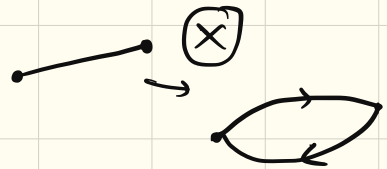
1
0 } check

$\Leftarrow$ $\longrightarrow$ By construction

D HAMILTONIAN PATH

$\downarrow$

U HAMILTONIAN PATH



Conclusion

- Other models : Transformers, Boolean Circuits, Neural networks

Algorithms ~ upper bounds

Complexity theory ~ lower bounds

Any important problem you want to solve will likely be NP-complete

→ heuristics, approximations

→ Quantum Computing course.