$$\text{HITTINGSTRING} = \{<S> \mid S = \{s_1, s_2, s_3, \ldots, s_k\} \text{ of equallength strings over } \Sigma = \{0, 1, *\} \text{ where } |s_1| = |s_2| = |s_3| = \ldots = |s_k| = n$$

$<S>$ is in HITTINGSTRING if there exists a string $y \in \{0,1\}^n$ such that for each $s_t \in S$ there is some i, $1 \leq i \leq n$, for which the i^{th} character of s_t and the i^{th} character of y are identical.
EXAMPLE:

- $S = \{0*1, 100, 110, **0\}$
  $y = 010$ is a hitting string for S. Therefore $<S> \in$ HITTINGSTRING.

s_t	$\boxed{0}*1$	$10\boxed{0}$	$1\boxed{1}0$	$**\boxed{0}$
y	$\boxed{0}10$	$01\boxed{0}$	$0\boxed{1}0$	$01\boxed{0}$

- $S = \{*01, 111, **0, 011\}$ does not allow any hitting string.
  So $<S> \notin$ HITTINGSTRING.

Claim: HITTINGSTRING is $\mathcal{NP}$–COMPLETE.
Proof:

- HITTINGSTRING $\in \mathcal{NP}$.
  y is a certificate and it takes polynomial time to check **and** verify.

- HITTINGSTRING is $\mathcal{NP}$–HARD.

$$\text{SAT} \leq_p \text{HITTINGSTRING}$$

Given $\Phi = \Phi_1 \wedge \Phi_2 \wedge \Phi_3 \wedge \ldots \wedge \Phi_k$ with n different variables, we want to construct a set S_1 of size k of n–long strings over $\Sigma = \{0, 1, *\}$. To each clause Φ_p we associate an n–long string s_p according to the following rule:

x_i	$\overline{x_i}$	i^{th} symbol of s_p
$\in \Phi_p$	$\in \Phi p$	$*$
$\in \Phi_p$	$\notin \Phi p$	1
$\notin \Phi_p$	$\in \Phi p$	0
$\notin \Phi_p$	$\notin \Phi p$	$*$

EXAMPLE: $\Phi = (x_1 \vee x_2 \vee \overline{x_4}) \wedge \overline{x_3} \wedge (x_1 \vee \overline{x_2} \vee x_3 \vee x_2)$
$n = 4$, $k = 3$

$$\begin{array}{rl} \Phi_1 = x_1 x_2 \overline{x_4} & s_1 = 11*0 \\ \Phi_2 = \overline{x_3} & s_2 = *0** \\ \Phi_3 = (x_1 \overline{x_2} x_3 x_2) & s_3 = 1*1* \end{array}$$

So $S = \{11{*}0, {*}0{*}{*}, 1{*}1{*}\}$. Note that Φ is satisfiable; a satisfiable assignment would be: $x_1 = x_3 = 1$ and $x_2 = x_4 = 0$. Note $y = 1010$ is a hitting string for S. The conversion technique (basically a table lookup) is done in polynomial-time (construction of all s_i's takes $O(kn)$ time).

$$\Phi \in \text{SAT} \Leftrightarrow <S> \in \text{HITTINGSTRING}$$

$(\Rightarrow)$: Given $\Phi = \Phi_1 \wedge \Phi_2 \wedge \Phi_3 \wedge \ldots \wedge \Phi_k$, construct $<S>$ by the outlined method above.

$$\begin{array}{rll} \Phi \in \text{SAT} & \Rightarrow & \exists \text{ a satisfying assignment } A = \{0,1\}^* \\ & \Rightarrow & \text{each } \Phi_p \text{ has at least one literal, which is true} \\ & \Rightarrow & A \text{ is a hitting string for } <S>. \end{array}$$

Or more formally, construct a hitting string y according to the following rule:

- $*$ put 1 in i^{th} position of y if x_i is true in A.
- $*$ put 0 in i^{th} position of y if x_i is false in A.

Then y is a hitting string for S.

$(\Leftarrow)$: Given some set $<S>$ with $<S> \in$ HITTINGSTRING, the following holds:

$$< S >\in \text{ HITTINGSTRING } \Rightarrow \exists\, y \in \{0,1\}^{ast} \text{ such that } y \text{ is a hitting string of } <S> \text{ and } |y| = n$$

Since each string in $<S>$ corresponds to a clause Φ_p in Φ, then at least one literal is the one that corresponds to a "hit" in $<S>$.

$$\Rightarrow \Phi \text{ is satisfiable}$$
$$\Rightarrow \Phi \in \text{SAT}$$

HALF–CLIQUE–Problem

$$\text{HALF–CLIQUE} = \{<G> \mid G \text{ is a graph with a clique of size } \left\lceil \tfrac{|V|}{2} \right\rceil\}$$

Claim: HALF–CLIQUE is $\mathcal{NP}$–COMPLETE
Proof:

- HALF–CLIQUE $\in \mathcal{NP}$.
  Use the set of $\lceil |V|/2 \rceil$ vertices in the clique as certificate of G. Checking whether it is a clique is accomplished in polynomial time by checking whether, for every pair u, v in the set, the edge $(u, v) \in E$.

- HALF–CLIQUE $\in \mathcal{NP}$–HARD.
  Show that

  $$\text{CLIQUE} \leq_p \text{HALF–CLIQUE}$$

  The reduction takes a graph $G(V, E)$ and

  $$<G, k> \in \text{CLIQUE} \Leftrightarrow <G'> \in \text{HALF–CLIQUE}$$

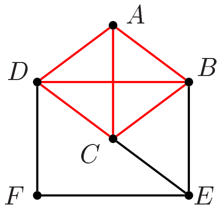
  The reduction depends on k:

  - if $k = \left\lceil \frac{|V|}{2} \right\rceil$, then let $G' = G$
  - if $k > \left\lceil \frac{|V|}{2} \right\rceil$, add m vertices such that $k = \left\lceil \frac{|V|}{2} \right\rceil$, then let $G' = G \cup \{q_1, q_2, \ldots, q_m\}$ where the newly added nodes $q_1, q_2, \ldots, q_m$ are of degree 0 (they are not adjacent to any nodes). Now $k = \left\lceil \frac{|V|}{2} \right\rceil$ and

  $$G \text{ has a clique of size } k \Leftrightarrow G' \text{ has a clique of size } \left\lceil \tfrac{|V|}{2} \right\rceil$$

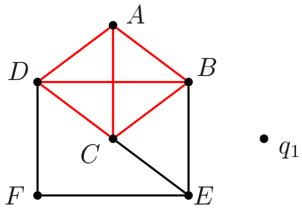
  Example:

  $G = (V, E)$
  $k = 4$
  $\left\lceil \frac{|V|}{2} \right\rceil = 3$



$\{A, B, C, D\}$ is a clique of size 4

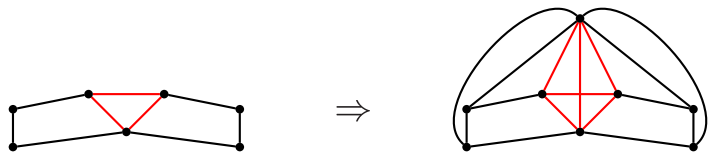
$\Rightarrow$



$G' = (V', E)$, $\left\lceil \frac{|V|}{2} \right\rceil = 4$
$V' = V \cup \{q_1\}$

- if $k < \left\lceil \frac{|V|}{2} \right\rceil$, add m vertices such that $k + m = \left\lceil \frac{|V|}{2} \right\rceil$ then let $G' = G \cup \{q_1, q_2, \ldots, q_m\}$ where the newly added vertices $q_1, q_2, \ldots, q_m$ are adjacent to all nodes of G and every q_1 is adjacent to q_j, $i \neq j$.

  EXAMPLE:



Clique of size 3 | Clique of size 4

So now $|V'| = |V| + m$ and

$$G \text{ has a clique of size } k \Leftrightarrow G' \text{ has a clique of size } k + m$$

Note that $k + m = \lceil |V'|/2 \rceil$ and $|V'| = |V| + m$, hence $2k + 2m = |V'|$ and so $2k + 2m = |V| + m$. Therefore $|V| = 2k + m$. In all cases, the "$\Leftrightarrow$" is satisfied. The reduction is in polyonmial time. So

$$\text{CLIQUE} \leq_p \text{HALF–CLIQUE}$$
$$\Rightarrow \text{HALF–CLIQUE} \in \mathcal{NP}\text{–COMPLETE}$$

A hamiltonian path in a graph $G = (V, E)$ is a path that traverses each vertex exactly once. More precisely:

HAM-PATH = $\{<G, u, v> \mid \exists$ a hamiltonian path from u to v in graph $G\}$

Claim: HAM–PATH $\in \mathcal{NP}$–COMPLETE

Proof:

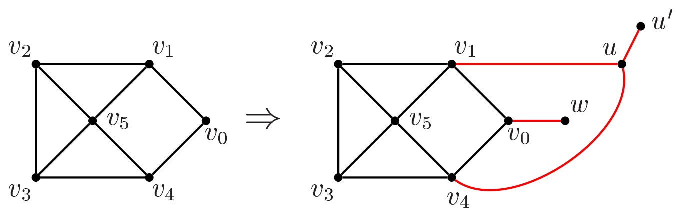
- HAMP–PATH $\in \mathcal{NP}$ since a path can serve as a certificate and be checked in polynomial time to be hamiltonian.
- HAM–PATH $\in \mathcal{NP}$–HARD.

  Claim: HAM–CYCLE $\leq_p$ HAM–PATH

  Transform any graph $G = (V, E)$ into $G' = (V', E)$ where

  a) $V' = V \cup \{u, u', w\}$
  b) $E' = E \cup \{(u', u), (w, v_0)\} \cup \{(u, v) | (v_0, v) \in E\}$ for some $v_0 \in E$.

  Adding nodes and check is achieved in polynomial time.



$$G \text{ has a hamiltoninan cycle} \Leftrightarrow G' \text{ has a hamiltonian path}$$

($\Leftarrow$) Suppose G' has a hamiltonian path P. The extreme edges of P have to be (u', u) and (v_0, w). Suppose $(u, v) \in P$. The rest of P is a path traversing each node in $V - \{v_0, v\}$ exactly once. Call it C.

$$(u, v) \in E' \;\Rightarrow\; (v, v_0) \in E$$

So C with (v_0, v) is a hamiltonian cycle in G.

($\Rightarrow$) If G has a hamiltonian cycle $C = \{v_0, \ldots, v, v_0\}$ then G' will have the hamiltonian path $P = [w, v_0, \ldots, v, u, u']$.

$$\text{SPATH} = \{<G, a, b, k> \mid G \text{ is a graph containing a simpel path of length at most } k \text{ from } a \text{ to } b\ \}.$$

Claim: SPATH $\in \mathcal{P}$
Proof: The following is a breadth–first search algorithm which runs in polynomial time.

```
SPATH(G, a, b, k)
begin
  mark a
  repeat k times or until b is marked
  (whichever takes fewer iterations)
    for each marked node n not yet
    expanded, expand it
  accept if b is marked and reject otherwise
end
```

To expand a node means to mark all nodes ajdacent to it. Also recall: u and v are adjacent nodes if $(u, v) \in E$.

$$\text{LPATH} = \{<G, a, b, k> \mid G \text{ is a graph containing a simpel path of length at least } k \text{ from } a \text{ to } b\ \}.$$

Claim: LPATH $\in \mathcal{NP}$–COMPLETE.
Proof:

- Given a set of vertices, we can check in polynomial time whether it is a simple path of length at least k.
- To show that LPATH $\in \mathcal{NP}$–HARD, we reduce HAM–PATH to it in polynomial time. That is, we show

  $$\text{HAM–PATH} \leq_p \text{LPATH}$$

  Let $(G, a, b) \in$ HAM–PATH, where $G = (V, E)$. (G, a, b) can be polynomial time reduced to $(G, a, b, n - 1)$, where $n = |V|$. Since there are only n vertices in G, a simple path (which cannot repeat vertices) of length exactly $n - 1$ is a hamiltonian path and vice versa.