§5 Intermission: Complexity Theory

Def: binary *length* of $x \in \mathbb{N}$ is $\ell(x) := \lceil \log_2(x+1) \rceil + 1$

- time=#steps of an algorithm on input $\underline{x}=(x_1,\ldots x_k)$
- space (=memory): $\max_t \ell(y_{1,t},.. y_{m,t})$, $\underline{y}$ registers
- worst-case over all inputs $\underline{x}$ with $\ell(\underline{x})<n$

Def: For decision problems $L \subseteq \mathbb{N}^*$ or $\subseteq \{0,1\}^*$

- $\mathcal{P}$ = { L decidable in polynomial time }
- $\mathcal{NP}$ = { L verifiable in polynomial time }, i.e.

$L = \{ \underline{x} \in \mathbb{N}^* : \exists \underline{y} \in \mathbb{N}^*, \ell(\underline{y}) \leq \text{poly}(\ell(\underline{x})), (\underline{x},\underline{y}) \in V \}, V \in \mathcal{P}$

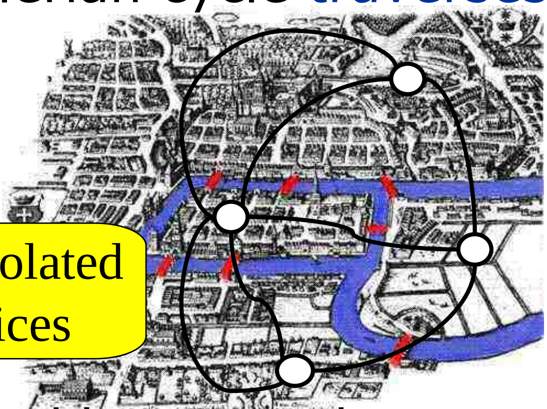
- $\mathcal{PSPACE}$ = { L decidable in polynomial space }
- $\mathcal{EXP}$ = { L decidable in exponential time }

Theorem: $\mathcal{P} \subseteq \mathcal{NP} \subseteq \mathcal{PSPACE} \subseteq \mathcal{EXP}$

Example Decision Problems I

In an undirected graph *G*, Eulerian cycle traverses each <u>edge</u> precisely once;

Hamiltonian cycle visits each <u>vertex</u> precisely once.



G admitting a Eulerian cycle is connected and has an even number of edges incident to each vertex

save isolated vertices

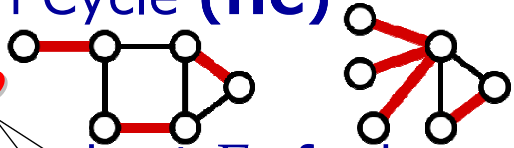
Theorem: Conversely every connected graph with an even number of edges incident to each vertex admits a Eulerian cycle.

EC := { $\langle G \rangle$ | G has a Eulerian cycle} $\mathcal{NP}$

HC := { $\langle G \rangle$ | G has Hamiltonian cycle} $\mathcal{NP}$

Example Decision Problems II

- Eulerian **(EC)** vs. Hamiltonian Cycle **(HC)**

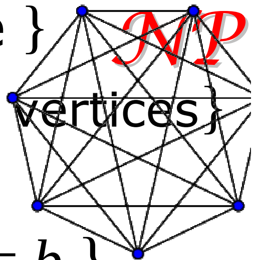


- (Minimum) **Edge Cover** $\mathcal{NP}$

  "To graph G, find a smallest subset F of edges s.t. any vertex v is adjacent to at least one $e \in F$."

  Greedily extend a maximum matching

- vs. **Vertex Cover (VC)** $\mathcal{NP}$
- **CLIQUE** = $\{ \langle G,k \rangle \mid G$ contains a k-clique $\}$ $\mathcal{NP}$
- **IS**=$\{\langle G,k\rangle : G$ has k pairwise non-adjacent vertices$\}$



- Integer Linear Programming $\mathcal{NP}$?

ILP = $\{ \langle \underline{A},\underline{b} \rangle : \underline{A} \in \mathbb{Z}^{n\times m}, \underline{b} \in \mathbb{Z}^m, \exists \underline{x} \in \mathbb{Z}^n : \underline{A}\cdot\underline{x} = \underline{b} \}$

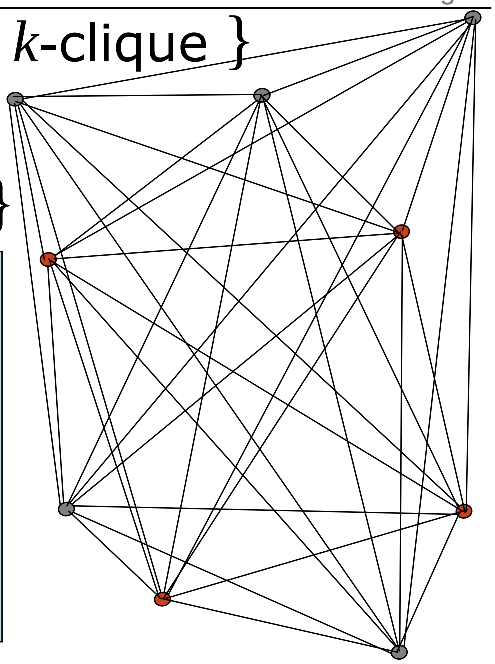
VC = $\{\langle V,E,k\rangle : \exists U \subseteq V, |U|=k, \forall (x,y)\in E: x\in U \vee y \in U \}$

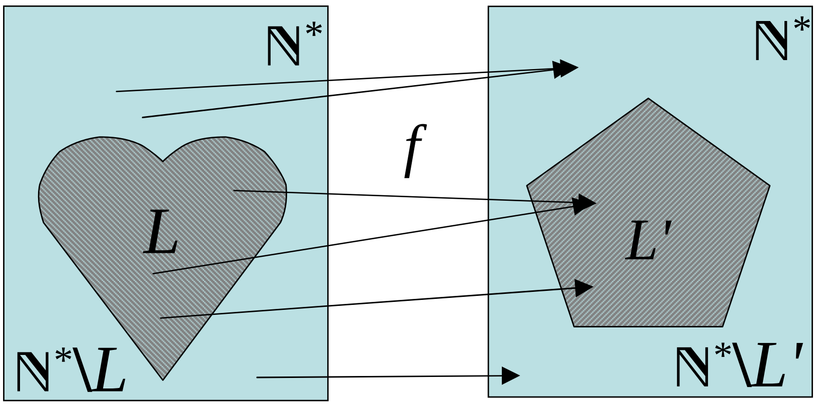
$\mathcal{NP} \ni \{x\in\mathbb{N} : \exists y, \ell(y) \leq \text{poly}(\ell(x)), \langle x,y\rangle \in V \}, \; V \in \mathcal{P}$

Comparing Decision Problems

CLIQUE = $\{ \langle G,k \rangle \mid G$ contains a k-clique $\}$

$\equiv_p$ **IS**=$\{ \langle G,k\rangle : G$ has k pairwise non-connected vertices$\}$





For $L,L' \subseteq \mathbb{N}^*$ write $L \preccurlyeq_p L'$ if exists a polynomial-time computable $f:\mathbb{N}^* \to \mathbb{N}^*$ such that $\forall \underline{x}: \; \underline{x}\in L \Leftrightarrow f(\underline{x}) \in L'$

Lemma: a) $L' \in \mathcal{P} \Rightarrow L \in \mathcal{P}$ b) $L \preccurlyeq_p L' \preccurlyeq_p L'' \Rightarrow L \preccurlyeq_p L''$

Reduction IS $\leqslant_p$ SAT

Goal: Upon input of (the encoding of) a graph G and $k \in \mathbb{N}$, produce in polynomial time a CNF formula Φ such that:
Φ satisfiable iff G contains $\geq k$ independent vertices

Let G consist of vertices $V=\{1,..,n\}$ and edges E.

- Consider Boolean variables $x_{v,i}$, $v \in V$, $i=1...k$
  Vertex *v* is #*i* among the *k* independent.
- and clauses $K_i := \bigvee_{v\in V} x_{v,i}$, $i=1...k$ — There is an *i*-th vertex
- and $\neg x_{v,i} \vee \neg x_{v,j}$, $v \in V$, $1 \leq i < j \leq k$ — Vertex *v* cannot be both #*i* and #*j*.
- and $\neg x_{u,i} \vee \neg x_{v,j}$, $\{u,v\} \in E$, $1 \leq i < j \leq k$ — No adjacent vertices are independent.
- Length of Φ: $O(k \cdot n + n \cdot k^2 + n^2 k^2) = O(n^2 k^2)$ since $k \leq n$.
- Computational cost of $(G,k) \to \Phi$: polyn. in $n + \log k$

Example Reduction: 4SAT vs. 3SAT

4-SAT: Is formula $\Phi(\underline{Y})$ in 4-CNF satisfiable?
3-SAT: Is formula $\Phi(\underline{Y})$ in 3-CNF satisfiable?

Given $\Phi = (a \vee b \vee c \vee d) \wedge (p \vee q \vee r \vee s) \wedge \ldots$

with literals $a,b,c,d,p,q,r,s,\ldots$ (variables, possibly negated)

Introduce new variables $u,v,\ldots$ and consider

$$\Phi' := (a \vee b \vee u) \wedge (\neg u \vee c \vee d) \wedge (p \vee q \vee v) \wedge (\neg v \vee \vee r \vee s) \wedge \ldots$$

$f: \langle\Phi\rangle \to \langle\Phi'\rangle$

For $L,L' \subseteq \mathbb{N}$ write $L \leqslant L'$ if exists a computable $f: \mathbb{N} \to \mathbb{N}$ such that $\forall \underline{x}$: $\underline{x} \in L \Leftrightarrow f(\underline{x}) \in L'$.