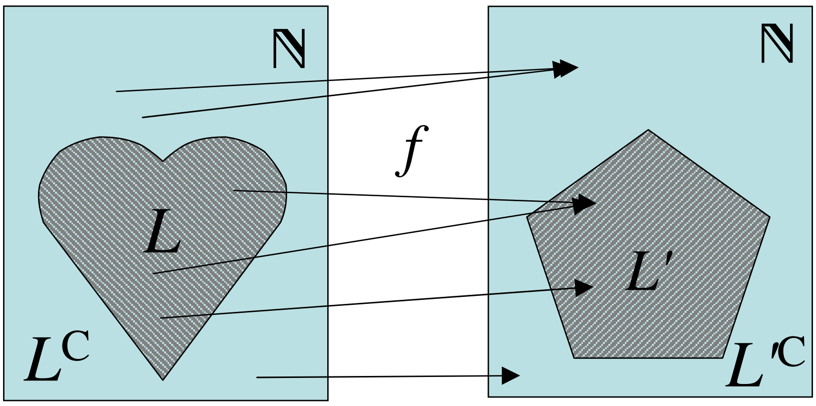
§5 Structural Complexity: $\mathcal{NP}c$

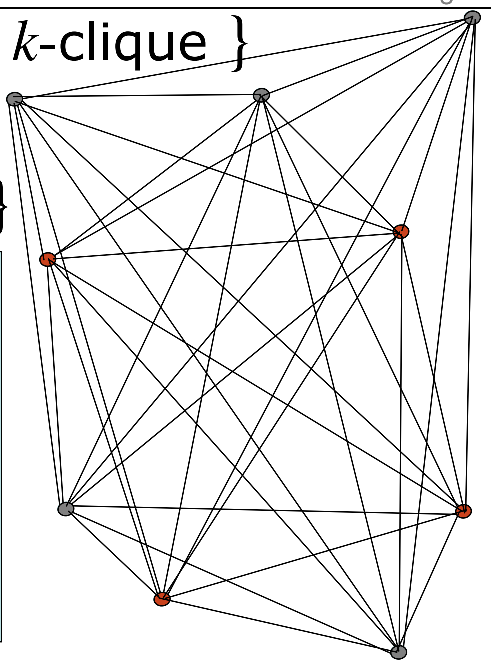
- Comparing Problems:
  polynomial-time reduction
- Reductions:
  CLIQUE $\equiv_p$ **IS** $\preccurlyeq_p$ **SAT** $\equiv_p$ **3SAT** $\preccurlyeq_p$ **IS**
- $\mathcal{NP}$-completeness, Master Reduction
- $co\mathcal{NP}$, Ladner's Theorem

Comparing Problems, again

CLIQUE = { $\langle G,k\rangle$ | 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 $\forall \underline{x}$: $\underline{x}\in L \Leftrightarrow f(\underline{x})\in L'$.

a) $L'\in\mathcal{P} \Rightarrow L\in\mathcal{P}$ / L' decidable $\Rightarrow$ so L. b) $L \preccurlyeq_p L' \preccurlyeq_p L'' \Rightarrow L\preccurlyeq_p L''$

Reduction IS $\preccurlyeq_p$ SAT

KAIST
CS422 M. Ziegler

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

KAIST
CS422 M. Ziegler

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

Reduction 3SAT $\preccurlyeq_p$ IS

KAIST

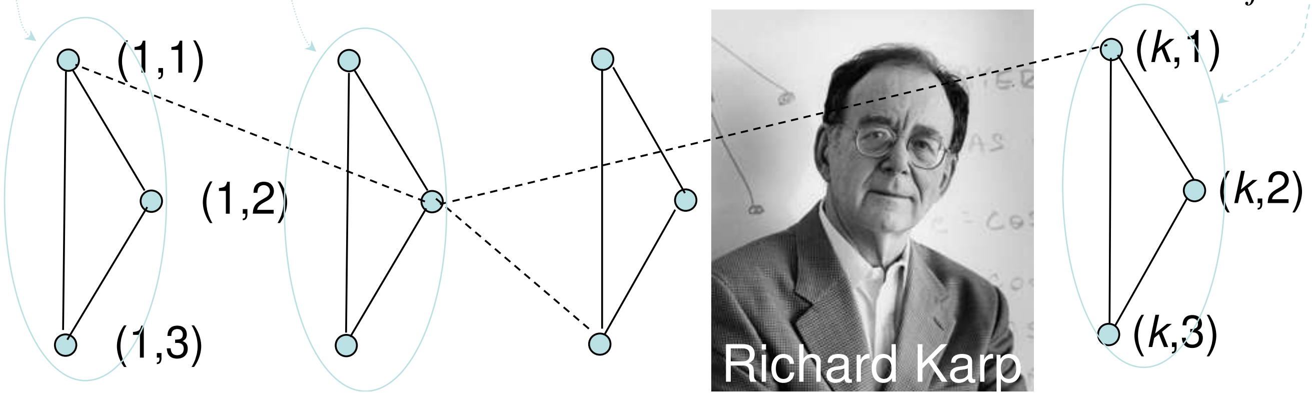
CS422 M. Ziegler

Produce, given a 3-CNF term Φ, within polynomial time a graph G and integer k such that it holds: Φ *is satisfiable iff* G *contains* k *pairwise non-adjacent vertices.*

e.g. $(u \vee .. \vee ..) \wedge (.. \vee \neg u \vee ..) \wedge (.. \vee .. \vee u) \wedge (u \vee .. \vee ..)$

$\Phi = C_1 \wedge C_2 \ldots \wedge C_k$, $C_i = x_{i1} \vee x_{i2} \vee x_{i3}$, x_{is} literals

$V := \{ (i,1), \ldots (i,3) : i \leq k \}$, $E := \{ \{(i,s),(j,t)\} : i=j \text{ or } \overline{x}_{is} = x_{jt} \}$



Problems of similar complexity

KAIST

CS422 M. Ziegler

unknown yet

- Showed: **CLIQUE** $\equiv_p$ **IS** $\preccurlyeq_p$ **SAT** $\equiv_p$ **3SAT** $\preccurlyeq_p$ **IS**.
- These 4 problem have about same complexity:
  - Either all are belong to $\mathcal{P}$, or none of them.
- We will show: Also **TSP**, **HC**, **VC** and many further problems in $\mathcal{NP}$ belong to this class called $\mathcal{NP}$c.
- **And** will show: These are 'hardest' problems in $\mathcal{NP}$.

Cook–Levin Theorem: $\forall L \in \mathcal{NP} : L \preccurlyeq_p$ **SAT**.

- That is, either *all* or *none* of the problems in $\mathcal{NP}$c can be *decided* in polynomial time.

In the first case:

- A deterministic **`WHILE+`** program could simulate any *non*-deterministic one with polynomial slowdown!

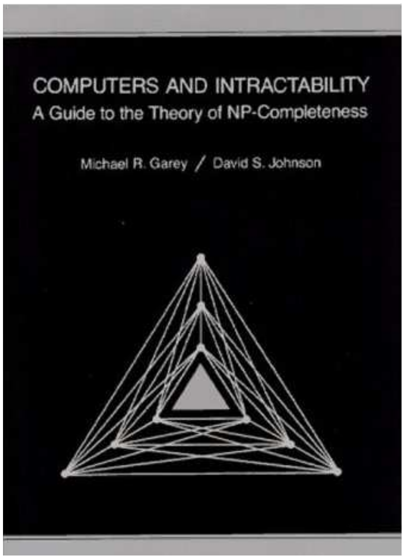
Complexity Class Picture

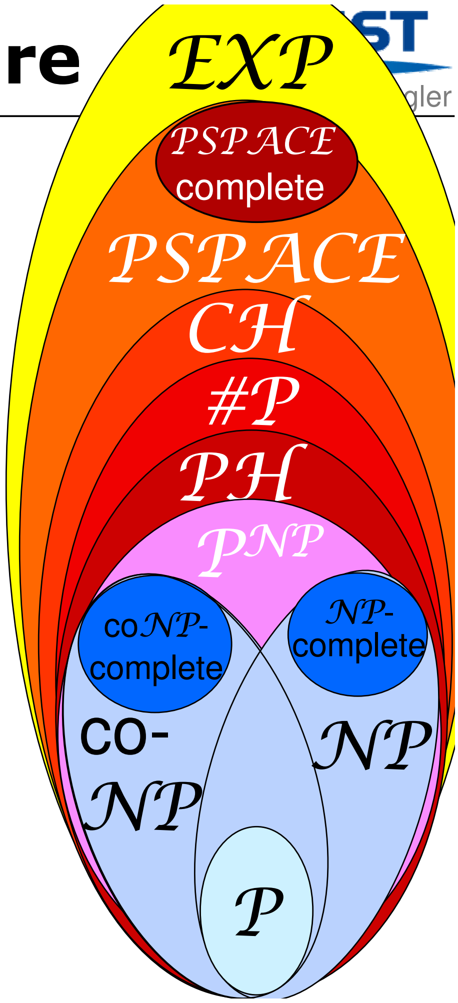
Def: $A \in \mathcal{NP}$ is $\mathcal{NP}$-complete if $L \preccurlyeq_p A$ holds for every $L \in \mathcal{NP}$.

Theorem (Cook'72/Levin'71): **SAT** is $\mathcal{NP}$-complete!

Lemma: For A $\mathcal{NP}$-complete and $A \preccurlyeq_p B \in \mathcal{NP}$, B is also $\mathcal{NPc}$.

Now know ≈500 natural problems $\mathcal{NP}$-complete…



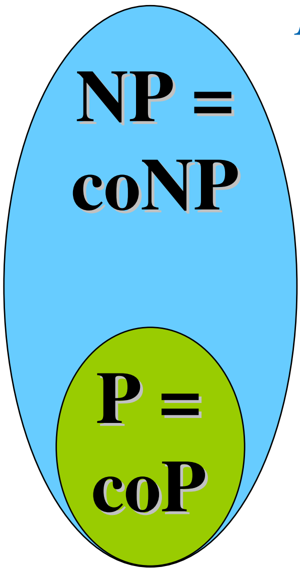


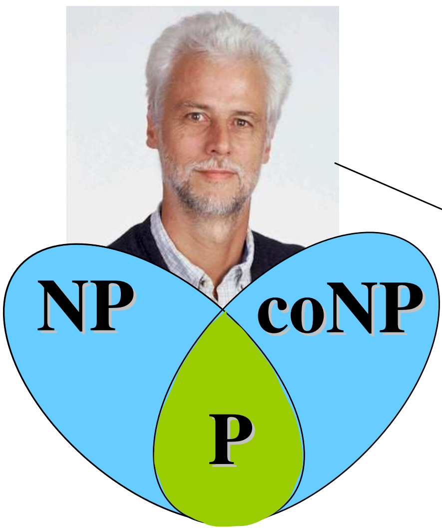
Scenarios for $\mathcal{P} \neq \mathcal{NP}$

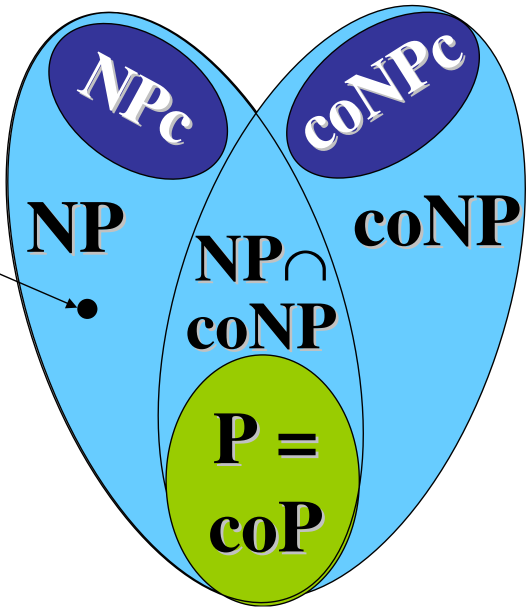
coNP := { $L \subseteq \mathbb{N}$: $\mathbb{N} \setminus L \in$ **NP** }

unSAT := { φ Boolean term, $\forall \underline{x}$: $\varphi(\underline{x})=0$ } ∈ **coNPc**

Theorem [Ladner'75]: If **P≠NP**, there exists $L \in$ **NP** \ (**P**∪**NPc**)







Master Reductions

SAT = { ⟨Φ⟩ : Φ Boolean term, $\exists y_1,\ldots y_m$: $\Phi(y_1,\ldots y_m)=1$ }

Cook/Levin Theorem: **SAT** is $\mathcal{NP}$-complete!

Proof (Sketch): Fix $L\in\mathcal{NP}$ and $V\in\mathcal{P}$.

Fix **WHILE+** program $\mathcal{B}$ deciding V in time poly(n).

Express "bin$(z_0,\ldots z_{k-1})\in V$" as Boolean term "$\Psi_k(z_0,\ldots z_{k-1})=1$" of length poly($k$).

$\mathbb{N}\ni x \to \Phi := \Psi_{n+k}(\text{bin}(x),\cdot)$

Then bin$(x_0,\ldots x_{n-1})\in L \Leftrightarrow \exists y_0,\ldots y_{m-1}\in\{0,1\}$: $\Psi_{n+k}(\underline{x},\underline{y})=1$

Thm: The following problem **UNP** is $\mathcal{NP}$-complete:
{ $\langle\mathcal{A},x,2^N\rangle$: nondetermin. **WHILE+** program $\mathcal{A}$ accepts input x within at most N steps }

$L = \{x\in\mathbb{N}: \exists y,\ \ell(y)\le\text{poly}(\ell(x)),\ \langle x,y\rangle\in V\},\ V\in\mathcal{P}$

SubsetSum is $\mathcal{NP}$-complete

{ $\langle a_1,\ldots a_N,b\rangle \mid a_1,\ldots a_N,b\in\mathbb{N},\ \exists\alpha_1,\ldots\alpha_N\in\{0,1\}: b=\sum_i a_i\cdot\alpha_i$ }

- **SubsetSum**$\in\mathcal{NP}$ √ Show: **3SAT** $\preccurlyeq_p$ **SubsetSum**
- In polyn.time: 3CNF $\Phi\to A\subseteq\mathbb{N}$ and $b\in\mathbb{N}$ s.t.
- ∃satisf. assignm. of $\Phi \Leftrightarrow \exists B\subseteq A$: $b=\sum_{a\in B} a$

Eg. $\Phi = (x_1\vee\neg x_3\vee x_5)\wedge(\neg x_1\vee x_5\vee x_4)\wedge(\neg x_2\vee\neg x_2\vee\neg x_5)$

v_1 :=	100 10000	v_1' :=	010 10000	b :=	444 11111
v_2 :=	000 01000	v_2' :=	002 01000	c_1 :=	100 00000
v_3 :=	000 00100	v_3' :=	100 00100	d_1 :=	200 00000
v_4 :=	010 00010	v_4' :=	000 00010	c_2 :=	010 00000
v_5 :=	110 00001	v_5' :=	001 00001	d_2 :=	020 00000
				c_3 :=	001 00000

m clauses in n var.s → $2n+2m+1$ values à $n+m$ dec.digits