NP search problem: \( \approx \text{NP-relation} \)
- goal: instance \( \rightarrow \) solution (any)
- for each instance, set of (valid/feasible) solutions
- can recognize instances & their solutions in P

- every NP problem \( \rightarrow \) NP search problem
  (for every choice of YES certificates \( \rightarrow \) solutions)

Counting version \( \#A \) of NP search problem A
- count number of solutions for given instance
- e.g. \( \#\text{SAT} \): find \# satisfying assignments
  \( \#\text{Shakashaka} \): find \# solutions to puzzle

\[ \#P = \{ \#A \mid \text{NP search problem } A \} \]
\[ = \{ \text{problems solved by polynomial-time nondeterministic counting algorithms} \} \]
\[ \rightarrow \text{makes guesses, at end says YES or NO} \]
\[ \text{(just like an NP algorithm)} \]
\[ \Rightarrow \text{output} = \#\text{guess paths leading to YES} \]

\( \#P \)-hard = as hard as all problems in \( \#P \)
- via multicolor (Cook-style) reductions
\[ \Rightarrow \&P \text{ unless } P = \text{NP} \]
- technically, \( \&P = \text{poly-time computable functions} \)
Parsimonious reduction for NP search problems

- instance $x$ of $A$ $\mapsto$ instance $x'$ of $B$
- computable in polynomial time (like NP reduction)
- $\#A$ solutions to $x = \#B$ solutions to $y$  
  $\Rightarrow$ decision problems (Is solution?) same answer
  $\Rightarrow$ NP reduction too

- $\#A$ is $\#P$-hard $\Rightarrow$ $\#B$ is $\#P$-hard

C-monious reduction: uniform scaling
- $c \cdot \#A$ solutions to $x = \#B$ solutions to $y$
- preserves $O \Rightarrow$ NP reduction too
- $\#A$ is $\#P$-hard $\Rightarrow$ $\#B$ is $\#P$-hard

$\#P$-complete SAT problems:
- $\#3SAT$
- planar $\#3SAT$
- planar monotone rectilinear $\#3SAT$
- planar positive rectilinear $\#1-in-3SAT$
- planar positive $\#2SAT-3$

$\{\text{as in L7}\}$

- Schaefer-style dichotomy:
  - $\#SAT \in FP \iff$ system of linear equations (mod 2)
  - $\#SAT$ $\#P$-complete otherwise

$[\text{Creignou & Hermann-I&CF 2006]}$

see $[\text{Creignou, Khanna, Sudan-SIGACT 2001}]$
Shakashaka: parsimonious $\Rightarrow$ \#P-hard
[Demaine, Okamoto, Uehara, Uno - CCCG 2013]

Hamiltonian cycles:
- old proofs not parsimonious [Lichtenstein] [Plesnik]
- parsimonious reduction from 3SAT to planar max-degree-3 Hamiltonian cycle [Sato - senior thesis 2002]
- nonplanar case solved earlier [Valiant 1974]

Slitherlink: parsimonious $\Rightarrow$ \#P-hard [Yato 2000]
- here can't use grid graphs
  $\Rightarrow$ optional vertex gadgets
Determinant of $n \times n$ matrix $A = (a_{ij})$ ∈ P

$$= \sum_{\text{permutation } \pi} (-1)^{\text{sign}(\pi)} \prod_{i=1}^{n} a_{i \pi(i)}$$

product of permutation matrix within $A$

Permanent

$$= \sum_{\text{permutation } \pi} \prod_{i=1}^{n} a_{i \pi(i)}$$

$\Rightarrow$ weighted directed $n$-node graph $w(i,j) = a_{ij}$:

$$= \sum_{\text{product of edge weights}} \text{cycle cover}$$

vertex-disjoint directed cycles hitting all vertices

$\Rightarrow$ $\#P$-complete [Valiant-TCS 1979]

$\Rightarrow$ c-monious reduction from $\#3SAT$

$\Rightarrow$ weight-1 edges in variable & clause gadgets

$\Rightarrow$ special weight matrix $X$ in junctions

$\Rightarrow$ perm $X = 0$ $\Rightarrow$ not alone in nonzero cycle cover

$\Rightarrow$ entered & exited by bigger cycle

$\Rightarrow$ perm $(X - \text{row} \& \text{col. 1}) = \text{perm}(X - \text{row} \& \text{col. 4}) = 0$

$\Rightarrow$ can't enter & leave immediately

$\Rightarrow$ enter at one end (1 or 4), leave at other

$\Rightarrow$ perm $(X - \text{rows} \& \text{cols. 1} \& 4) = 0$

$\Rightarrow$ can't leave interior $2 \times 2$ separate

$\Rightarrow$ must be visited between enter & exit

$\Rightarrow$ perm $(X - \text{row 1} - \text{col. 4}) = \text{perm}(X - \text{row 4} - \text{col. 1}) = 4$

factor for each traversal

$\Rightarrow$ acts as forced edge in var. & clause gadgets

$\Rightarrow$ perm = $4^8 \cdot \#\text{clauses} \cdot \#\text{satisfying assignments}$
**Permanant mod r** also \#P-hard: [Valiant-TCS 1979]
- multical reduction from Permanant
- set \( r = 2, 3, 5, 7, 11, \ldots \) until product > \( M^n \cdot n! \)
largest absolute entry in matrix <
\[ \Rightarrow O(n \lg M + n \lg n) \text{ calls } \& \max r = O(\text{that ln that}) \]
- use Chinese Remainder theorem [Prime # theorem]

**0/1-permanent mod r:** [Valiant-TCS 1979]
- parsimonious reduction from permanent mod r
  \[ \Rightarrow \text{all edge weights (effectively) nonnegative} \]
- replace weight-k edge (k>1) with gadget with k loops
- unique solution if original edge unused
- exactly k solutions if original edge used
  (using exactly 1 loop)

**0/1-permanent:** [Valiant-TCS 1979]
- one-call reduction from 0/1-permanent mod r
- call with same input
- return output \( \text{ mod } r \)

\[ = \text{# cycle covers in given directed graph} \]
\[ = \text{# perfect matchings in given bipartite graph} \]
\[ (V_1 = \text{rows, } V_2 = \text{columns, } (i, j) \in E \Leftrightarrow a_{ij} = 1) \]
\[ V_1 \overrightarrow{V_2} \]
(balanced: \( |V_1| = |V_2| \))
Bipartite # maximal matchings: [Valiant – SICOMP 1977]
- one-call reduction from bipartite # perfect matchings
- replace each vertex with \( n \) copies \((n=1!v!1)\)
  & each edge with biclique \( K_{n,n} \)
  \( \Rightarrow \) old matching of size \( i \)
  \( \rightarrow (n!)^i \) distinct matchings of size \( n \cdot i \)
  (& preserves maximality)
- \# maximal matchings
  \[ \leq \sum_{i=0}^{n/2} (\text{# orig. maximal matchings size } i) \cdot (n!)^i \leq (n/2)! \text{\ e.g. } Kn_{n/2}, n/2 \]
  \( \Rightarrow \) can extract \# perfect matchings \((i = n/2)\)

Bipartite # matchings: [Valiant – SICOMP 1977]
- multicall reduction from bipartite # perfect matchings
- \( G \rightarrow G_k : \) for each vertex: add \( k \) adjacent leaves
- \( M_r \) matchings of size \( n/2-r \) in \( G \)
  contained in \( M_r (k+1)^r \) matchings in \( G_k \)
  \( \Rightarrow \) \# matchings in \( G_k = \sum_{r=0}^{n/2} M_r (k+1)^r \)
  - evaluate this polynomial for \( k = 1, 2, \ldots, n/2+1 \)
  \( \Rightarrow \) can extract coefficients \( M_0, M_1, \ldots \)
  - \( M_0 = \) desired \# perfect matchings in \( G \)
Positive \#2SAT
\[= \# \text{ vertex covers} \]
\[\Rightarrow \# \text{ cliques in complement graph} \]

- parsimonious reduction from bipartite \# matchings
- edge \(\rightarrow\) variable: true = not in the matching
- 2 incident edges e & f \(\rightarrow\) clause e \& f
\[\Rightarrow \text{satisfying assignment} = \text{matching} \]

\# Minimal Vertex Covers
\[= \# \text{ maximal cliques in complement graph} \]
\[= \# \text{ minimal truth settings for positive 2SAT} \]

- parsimonious reduction from bipartite
  \# maximal matchings, as above
- minimal satisfying assignment = maximal matching
\[|E|-i \text{ true variables} \leq \text{ size } i \]

3-regular bipartite planar \# Vertex Cover
\[= \text{ planar positive 2SAT-3} \]
\[\text{where each clause has 1 red & 1 blue variable} \]
\[-\#P\text{-complete} \quad [\text{Xia, Zhang, Zhao - TCS 2007}] \]

(2,3)-regular bipartite \# Perfect Matchings
\[-\#P\text{-complete} \quad [\text{Xia, Zhang, Zhao - TCS 2007}] \]

(note: decision versions easy)
Another Solution Problem (ASP) \cite{Ueda & Nagao - TR 1996}
- for NP search problem A:
  ASP A: given one solution, is there another?
  - useful in puzzle design: want unique solution
- e.g. ASP k-coloring $\in P$ (rotate colors)
  & ASP 3-regular Hamiltonian cycle $\in P$
  (always another solution)

ASP reduction: parsimonious reduction $A \rightarrow B$
  & poly.-time bijection between $solutions_A(x)$
  & $solutions_B(x')$
  - induces every parsimonious reduction we've seen
  $\Rightarrow$ ASP A $\rightarrow$ ASP B via NP reduction
  (can map given solution to A $\rightarrow$ sol. to B)
- ASP B $\in P$ $\Rightarrow$ ASP A $\in P$
- ASP A NP-hard $\Rightarrow$ ASP B NP-hard

ASP-hard $=$ ASP reducible from every NP search prob.
  $\Rightarrow$ NP-hard

ASP-complete $=$ ASP-hard NP search problem
  - includes planar 3SATs & Hamiltonicity today,
    Shakashaka, Slitherlink
  - not e-monius reductions: 2SAT, matchings, permanent