**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. \#SAT: find \# satisfying assignments
  \#Shakashaka: find \# solutions to puzzle

\[ \#P = \{ \#A \mid \text{NP search problem } A \}^3 \quad [\text{Valiant-TCS 1979}] \]
- problems solved by polynomial-time nondeterministic counting algorithms \( \exists \)
  - \( \exists \) makes guesses, at end says YES or NO (just like an NP algorithm)
  - output = \# guess paths leading to YES

\#P-hard = as hard as all problems in \#P
via multiclass (Cook-style) reductions

\( \Rightarrow \) \#P unless \( P = \text{NP} \)
  - technically, \#P = \text{poly-time computable functions}
Parsimonious reduction for NP search problems

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

- \( \#A \) is \( \#P \)-hard \( \Rightarrow \) \( \#B \) is \( \#P \)-hard
  \& \( A \) is \( NP \)-hard \( \Rightarrow \) \( B \) is \( NP \)-hard

C-monious reduction: uniform scaling
- \( \#A \) solutions to \( x = \#B \) solutions to \( y \)
- preserves \( \emptyset \) \( \Rightarrow \) 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 \( \#1 \text{-in-} 3SAT \)
- planar positive \( \#2SAT \text{-} 3 \)

\{ [Hunte II, Marathe, Radhakrishnan, Stearns - SICOMP 1998] \} as in \( L7 \)

Schaefer-style dichotomy:
- \( \#SAT \) \( \Leftrightarrow \) system of linear equations (mod 2)
- \( \#SAT \) \( \#P \)-complete otherwise

[Creignou & Hermann - I&C 2006]
see [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  
- here can’t use grid graphs  
  $\Rightarrow$ optional vertex gadgets  
  [Yato 2000]
Determinant of \( n \times n \) matrix \( A = (a_{ij}) \in \mathbb{P} \):

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

Product of permutation matrix within \( A \)

Permanent:

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

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

\[
\sum \text{product of edge weights } \text{ of cycle cover}
\]

Vertex-disjoint directed cycles hitting all vertices

- \#P-complete [Valiant, TCS 1979]
- \#P-monious reduction from \#3SAT
- Weight-1 edges in variable & clause gadgets
- Special weight matrix \( X \) in junctions
  - \( \text{perm } X = 0 \Rightarrow \) not alone in nonzero cycle cover
    ⇒ entered & exited by bigger cycle
  - \( \text{perm} (X - \text{row & col. } 1) = \text{perm} (X - \text{row & col. } 4) = 0 \)
    ⇒ can’t enter & leave immediately
    ⇒ enter at one end (1 or 4), leave at other
  - \( \text{perm} (X - \text{rows & cols. } 1 \& 4) = 0 \)
    ⇒ can’t leave interior 2x2 separate
    ⇒ must be visited between enter & exit
  - \( \text{perm} (X - \text{row } 1 - \text{col. } 4) = \text{perm} (X - \text{row } 4 - \text{col. } 1) = 4 \)
    Factor for each traversal

→ acts as forced edge in var. & clause gadgets

\( \Rightarrow \text{perm} = 4^8 \cdot \# \text{clauses} \cdot \# \text{satisfying assignments} \)
Permanent mod r also #P-hard: [Valiant–TCS 1979]
- multical reduction from Permanent
- set \( r = 2, 3, 5, 7, 11, \ldots \) until product \( > M^n \cdot n! \)
  largest absolute entry in matrix \( \leq r \)
\( \Rightarrow O(n \log M + n \log n) \) 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
  \( \Rightarrow \) 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 mod r

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

Bipartite # matchings: [Valiant - SICOMP 1977]
- multicall reduction from bipartite # perfect matchings
- G → 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
  ⇒ # matchings in G_k = \sum_{r=0}^{n/2} M_r (k+1)^r
  - evaluate this polynomial for k = 1, 2, ..., n/2+1
  ⇒ can extract coefficients M_0, M_1, ...
  - M_0 = desired # perfect matchings in G
Positive #2SAT

\[ \text{Valiant - SICOMP 1977} \]

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

# Minimal Vertex Covers

\[ \text{Valiant - SICOMP 1977} \]

- 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

- planar positive 2SAT-3
- where each clause has 1 red & 1 blue variable
- \#P-complete

(2,3)-regular bipartite #Perfect Matchings

- \#P-complete

(note: decision versions easy)
Another Solution Problem (ASP) [Ueda & Nagao - TR 1996]
- for NP search problem \( A \):
  \( \text{ASP} A \): given one solution, is there another?
- useful in puzzle design: want unique solution

- e.g. ASP \( k \)-coloring \( \in \text{P} \) (rotate colors)
  & ASP 3-regular Hamiltonian cycle \( \in \text{P} \)
  (always another solution)

\textbf{ASP reduction}: parsimonious reduction \( A \rightarrow B \)
& poly.-time bijection between solutions \( \text{solutions}_A(x) \)
& \( \text{solutions}_B(x') \)
- includes every parsimonious reduction we've seen
\( \Rightarrow \) \( \text{ASP} A \rightarrow \text{ASP} B \) via NP reduction
  (can map given solution to \( A \rightarrow \text{sol. to } B \))
- \( \text{ASP} B \in \text{P} \) \( \Rightarrow \) \( \text{ASP} A \in \text{P} \)
- \( \text{ASP} A \) NP-hard \( \Rightarrow \) \( \text{ASP} B \) NP-hard

\textbf{ASP-hard} = ASP reducible from every NP search prob.
\( \Rightarrow \) NP-hard

\textbf{ASP-complete} = ASP-hard NP search problem
- includes planar 3SATs & Hamiltonicity today,
  Shakashaka, Slitherlink
- not c-monius reductions: 2SAT, matchings, permanent