CYCLE COVER

• A set of node-disjoint cycles that cover all nodes in a directed graph is called a cycle cover.

• There are 3 cycle covers (in red) above.
CYCLE COVER and BIPARTITE PERFECT MATCHING

Proposition 96  CYCLE COVER and BIPARTITE PERFECT MATCHING (p. 532) are parsimoniously reducible to each other.

- A polynomial-time algorithm creates a bipartite graph $G'$ from any directed graph $G$ (see next page).
- The number of cycle covers for $G$ equals the number of bipartite perfect matchings for $G'$.
- And vice versa.

Corollary 97  CYCLE COVER $\in P$. 
Illustration of the Proof
Permanent

• The **permanent** of an $n \times n$ integer matrix $A$ is

$$\text{perm}(A) = \sum_{\pi} \prod_{i=1}^{n} A_{i,\pi(i)}.$$  

  – $\pi$ ranges over all permutations of $n$ elements.

• 0/1 PERMANENT computes the permanent of a 0/1 (binary) matrix.
  
  – The permanent of a binary matrix is at most $n!$.

• Simpler than determinant (9) on p. 536: no signs.

• Surprisingly, much harder to compute than determinant!
Permanent and Counting Perfect Matchings

- BIPARTITE PERFECT MATCHING is related to determinant.\(^a\)

- \(#\text{BIPARTITE PERFECT MATCHING}\) is related to permanent.

**Proposition 98** 0/1 PERMANENT and BIPARTITE PERFECT MATCHING are parsimoniously reducible to each other.

\(^{a}\)Recall p. 537.
The Proof

• Given a bipartite graph $G$, construct an $n \times n$ binary matrix $A$.
  - The $(i, j)$th entry $A_{ij}$ is 1 if $(i, j) \in E$ and 0 otherwise.

• Then $\text{perm}(A) = \text{number of perfect matchings in } G$. 
Illustration of the Proof Based on p. 872 (Left)

\[
A = \begin{bmatrix}
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1
\end{bmatrix}.
\]

- \(\text{perm}(A) = 4\).
- The permutation corresponding to the perfect matching on p. 872 is marked.
Permanent and Counting Cycle Covers

**Proposition 99** 0/1 PERMANENT and CYCLE COVER are parsimoniously reducible to each other.

- Let $A$ be the adjacency matrix of the graph on p. 872 (right).
- Then $\text{perm}(A) = \text{number of cycle covers}$. 
Three Parsimoniously Equivalent\(^a\) Problems

We summarize Propositions 96 (p. 871) and 98 (p. 874) in the following.

**Lemma 100** 0/1 PERMANENT, BIPARTITE PERFECT MATCHING, and CYCLE COVER are parsimoniously equivalent.

We will show that the counting versions of all three problems are in fact \#P-complete.

\(^a\)Meaning the numbers of solutions are equal in a reduction.
WEIGHTED CYCLE COVER

• Consider a directed graph $G$ with integer weights on the edges.

• The weight of a cycle cover is the product of its edge weights.

• The cycle count of $G$ is sum of the weights of all cycle covers.
  - Let $A$ be $G$’s adjacency matrix but $A_{ij} = w_i$ if the edge $(i, j)$ has weight $w_i$.
  - Then $\text{perm}(A) = G$’s cycle count (same proof as Proposition 99 on p. 877).

• \#CYCLE COVER is a special case: All weights are 1.
There are 3 cycle covers, and the cycle count is

\[(4 \cdot 1 \cdot 1) \cdot (1) + (1 \cdot 1) \cdot (2 \cdot 3) + (4 \cdot 2 \cdot 1 \cdot 1) = 18.\]

\[a\] Each edge has weight 1 unless stated otherwise.
Three \#P-Complete Counting Problems

**Theorem 101 (Valiant, 1979)** \(0/1\) PERMANENT, \#BIPARTITE PERFECT MATCHING, \textit{and} \#CYCLE COVER \textit{are} \#P-complete.

- By Lemma 100 (p. 878), it suffices to prove that \#CYCLE COVER \textit{is} \#P-complete.
- \#SAT \textit{is} \#P-complete (p. 868).
- \#3SAT \textit{is} \#P-complete because it and \#SAT \textit{are} parsimoniously equivalent.
- We shall prove that \#3SAT \textit{is} polynomial-time Turing-reducible to \#CYCLE COVER.
The Proof (continued)

- Let $\phi$ be the given 3SAT formula.
  - It contains $n$ variables and $m$ clauses (hence $3m$ literals).
  - It has $\#\phi$ satisfying truth assignments.
- First we construct a *weighted* directed graph $H$ with cycle count
  \[ \#H = 4^{3m} \times \#\phi. \]
- Then we construct an unweighted directed graph $G$.
- We shall make sure $\#H$ (hence $\#\phi$) is polynomial-time Turing-reducible to $\#G$ ($G$’s number of cycle covers).
The Proof: Comments (continued)

• Our reduction is not expected to be parsimonious.
  – Suppose otherwise and
    \[ \#\phi = \#G. \]
  – Hence \( G \) has a cycle cover if and only if \( \phi \) is satisfiable.
  – But cycle cover \( \in P \) (p. 871).
  – Thus 3SAT \( \in P \), a most unlikely event!
The Proof: the Clause Gadget (continued)

- Each clause is associated with a **clause gadget**.

![Diagram](attachment:image.png)

- Each edge has weight 1 unless stated otherwise.
- Each bold edge corresponds to one literal in the clause.
  - The assignment of literals to edges is arbitrary.
- Bold edges are schematic only, a not **parallel** edges.

\(^{a}\)Preview p. 897.
The Proof: the Clause Gadget (continued)

- Following a bold edge makes the literal false (0).
- A cycle cover cannot select all 3 bold edges.
  - The interior node would be missing.
- Every proper nonempty subset of bold edges corresponds to a unique cycle cover of weight 1 (see next page).
The Proof: the Clause Gadget (continued)

- 7 possible cycle covers, one for each satisfying assignment: (1) $a = 0, b = 0, c = 1$, (2) $a = 0, b = 1, c = 0$, etc.
The Proof: the XOR Gadget (continued)
The Proof: Properties of the XOR Gadget (continued)

- The XOR gadget schema:

\[ \begin{align*}
  & \quad u \\
  & \quad + \\
  & \quad v' \\
  & \quad + \\
  & \quad u' \\
  & \quad v
\end{align*} \]

- At most one of the 2 schematic edges will be included in a cycle cover.

- There are 3m XOR gadgets, one for each literal.

- Only those cycle covers that take exactly one schematic edge in every XOR gadget contribute a nonzero weight (see next pages).
The Proof: Properties of the XOR Gadget (continued)

- Total weight of $-1 - 2 + 6 - 3 = 0$ for cycle covers not entering or leaving it.
The Proof: Properties of the XOR Gadget (continued)

- Total weight of \(-1 + 1 - 6 + 2 + 3 + 1 = 0\) for cycle covers entering at \(u\) and leaving at \(v'\).\(^a\)

\[ u \quad v' \quad v \quad u' \]

- Same for cycle covers entering at \(v\) and leaving at \(u'\).

\(^a\)Corrected by Mr. Yu-Tsung Dai (B91201046) and Mr. Che-Wei Chang (R95922093) on December 27, 2006.
The Proof: Properties of the XOR Gadget (continued)

- Total weight of $1 + 2 + 2 - 1 + 1 - 1 = 4$ for cycle covers entering at $u$ and leaving at $u'$.

- Same for cycle covers entering at $v$ and leaving at $v'$.
The Proof: Summary (continued)

- Cycle covers not entering *all* of the XOR gadgets contribute 0 to the cycle count.
  - Let $x$ denote an XOR gadget not entered for some cycle covers for $H$.
  - Now, such cycle covers’ contribution to the cycle count totals, by p. 889,

\[
\sum_{\text{cycle cover } c \text{ not entering } x} \text{(weight of } c \text{ for } H) = \sum_{\text{cycle cover } c \text{ not entering } x} \text{(weight of } c \text{ for } H - x) \times \text{(weight of } c \text{ for } x) = \sum_{\text{cycle cover } c \text{ not entering } x} \text{(weight of } c \text{ for } H - x) \times 0 = 0.
\]
The Proof: Summary (continued)

- Cycle covers entering \textit{any} of the XOR gadgets and leaving illegally contribute 0 to the cycle count by p. 890.

- For every XOR gadget entered and exited legally, the total weight of a cycle cover is multiplied by 4.
  - Each such act multiplies the weight by 4 by p. 891.
The Proof: Summary (continued)

• Hereafter we consider only cycle covers which enter every XOR gadget and leaves it legally.
  – Only these cycle covers contribute nonzero weights to the cycle count.

• They are said to respect the XOR gadgets.
The Proof: the Choice Gadget (continued)

- One choice gadget (a schema) for each variable.

- It gives the truth assignment for the variable.

- Use it with the XOR gadget to enforce consistency.
Schema for \((w \lor x \lor \overline{y}) \land (\overline{x} \lor \overline{y} \lor \overline{z})\)
The Full Graph\(^a\) \((w \lor x \lor \bar{y}) \land (\bar{x} \lor \bar{y} \lor \bar{z})\)

\(^a\)The XOR gadgets (p. 887) are rotated here.
The Proof: a Key Observation (continued)

Each satisfying truth assignment to $\phi$ corresponds to a schematic cycle cover that respects the XOR gadgets.
\( w = 1, x = 0, y = 0, z = 1 \iff \text{One Cycle Cover} \)
The Proof: a Key Corollary (continued)

- Recall that there are $3m$ XOR gadgets.
- Each satisfying truth assignment to $\phi$ contributes $4^{3m}$ to the cycle count $\#H$.
- Hence

$$\#H = 4^{3m} \times \#\phi,$$

as desired.
“$w = 1, x = 0, y = 0, z = 1$” Adds $4^6$ to Cycle Count
The Proof (continued)

• We are almost done.

• The weighted directed graph $H$ needs to be efficiently replaced by an unweighted graph $G$.

• Furthermore, knowing $\#G$ should enable us to calculate $\#H$ efficiently.
  
  – This done, $\#\phi$ will have been Turing-reducible to $\#G$.

• We proceed to construct this graph $G$.

---

\(^a\)By way of $\#H$ of course.
The Proof: Construction of $G$ (continued)

- Replace edges of the XOR gadget (p. 887) with weights 2 and 3 without creating parallel edges:

  ![Diagram]

- The cycle count $\#H$ remains unchanged.
The Proof: Construction of $G$ (continued)

- We move on to edges with weight $-1$.
- First, we count the number of nodes, $M$.
- Each clause gadget (p. 884) contains 4 nodes, and there are $m$ of them (one per clause).
- Each choice gadget (p. 895) contains 2 nodes, and there are $n \leq 3m$ of them (one per variable).
- Each revised XOR gadget (p. 903) contains 7 nodes, and there are $3m$ of them (one per literal).
- So

\[
M \leq 4m + 6m + 21m = 31m.
\]
The Proof: Construction of $G$ (continued)

- $\#H \leq 2^L$ for some $L = O(m \log m)$.
  - The maximum absolute value of the edge weight is 1.
  - Hence each term in the permanent is at most 1.
  - There are $M! \leq (31m)!$ terms.
  - Hence

$$
\#H \leq \sqrt{2\pi(31m)} \left(\frac{31m}{e}\right)^{31m} e^{\frac{1}{12(31m)}}
= 2^{O(m \log m)} \quad (29)
$$

by a refined Stirling’s formula.
The Proof: Construction of $G$ (continued)

- Replace each edge with weight $-1$ with the following:

- Each increases the number of cycle covers $2^{L+1}$-fold.
- The desired unweighted $G$ has been obtained.
The Proof (continued)

• \( \#G \) equals \( \#H \) after replacing each appearance \(-1\) in \( \#H \) with \( 2^{L+1} \):

\[
\#H = \cdots + 1 \cdot 1 \cdots (\!-1\!) \cdots 1 + \cdots ,
\]

a cycle cover \((n\) terms\)

\[
\#G = \cdots + 1 \cdot 1 \cdots 2^{L+1} \cdots 1 + \cdots .
\]

a cycle cover \((n\) terms\)

• Let \( \#G = \sum_{i=0}^{n} a_i \times (2^{L+1})^i \), where \( 0 \leq a_i < 2^{L+1} \).

• Recall that \( \#H \leq 2^L \) (p. 905).

• So each \( a_i \) counts the number of cycle covers with \( i \) edges of weight \(-1\) as there is no “overflow” in \( \#G \).
The Proof (concluded)

- We conclude that
  \[
  \#H = a_0 - a_1 + a_2 - \cdots + (-1)^n a_n,
  \]
  indeed easily computable from \(\#G\).
- We know \(\#H = 4^{3m} \times \#\phi\) from Eq. (28) on p. 900.
- So
  \[
  \#\phi = \frac{a_0 - a_1 + a_2 - \cdots + (-1)^n a_n}{4^{3m}}.
  \]
  - Equivalently,
    \[
    \#\phi = \frac{\#G \mod (2^{L+1} + 1)}{4^{3m}}.
    \]
Finis