*Computation That Counts*

And though the holes were rather small,
they had to count them all.
— The Beatles, *A Day in the Life* (1967)

## Counting Problems

- Counting problems are concerned with the number of solutions.
  - #SAT: the number of satisfying truth assignments to a boolean formula.
  - #HAMILTONIAN PATH: the number of Hamiltonian paths in a graph.
- They cannot be easier than their decision versions.
  - The decision problem has a solution if and only if the solution count is larger than 0.
- But they can be harder than their decision versions.

## Decision and Counting Problems

- FP is the set of polynomial-time computable functions $f : \{ 0, 1 \}^* \to \mathbb{Z}$.
  - GCD, LCM, matrix-matrix multiplication, etc.
- If $\#\text{SAT} \in \text{FP}$, then $\text{P} = \text{NP}$.
  - Given boolean formula $\phi$, calculate its number of satisfying truth assignments, $k$, in polynomial time.
  - Declare "$\phi \in \text{SAT}$" if and only if $k \geq 1$.
- The validity of the reverse direction is open.

## A Counting Problem Harder than Its Decision Version

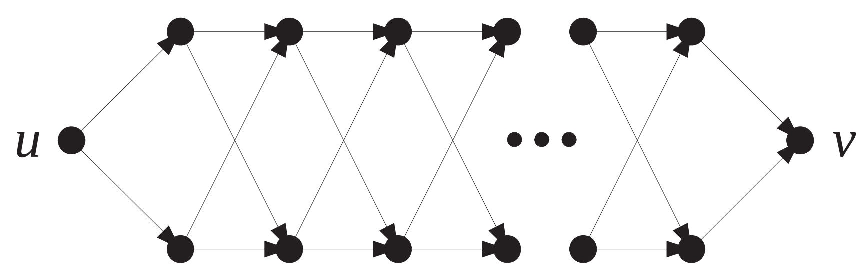
- CYCLE asks if a directed graph contains a cycle.[a]
- #CYCLE counts the number of cycles in a directed graph.
- CYCLE is in P by a simple greedy algorithm.
- But #CYCLE is hard unless P = NP.

[a] A cycle has no repeated nodes.

## Hardness of #CYCLE

**Theorem 97 (Arora, 2006)** *If #CYCLE $\in FP$, then $P = NP$.*

- It suffices to reduce the NP-complete HAMILTONIAN CYCLE to #CYCLE.
- Consider a *directed* graph $G$ with $n$ nodes.
- Define $N \equiv \lfloor n \log_2(n+1) \rfloor$.
- Replace each edge $(u, v) \in G$ with this subgraph:



## The Proof (continued)

- This subgraph has $N+1$ levels.
- There are now $2^N$ paths from $u$ to $v$.
- Call the resulting digraph $G'$.
- Recall that a Hamiltonian cycle on $G$ contains $n$ edges.
- To each Hamiltonian cycle on $G$, there correspond $(2^N)^n = 2^{nN}$ cycles (not necessarily Hamiltonian) on $G'$.
- So if $G$ contains a Hamiltonian cycle, then $G'$ contains at least $2^{nN}$ cycles.

## The Proof (continued)

- Now suppose $G$ contains no Hamiltonian cycles.
- Then every cycle on $G$ contains at most $n-1$ nodes.
- There are hence at most $n^{n-1}$ cycles on $G$.
- Each $k$-node cycle on $G$ induces $(2^N)^k$ cycles on $G'$.
- So $G'$ contains at most $n^{n-1}(2^N)^{n-1}$ cycles.
- As $n \geq 1$,

$$
\begin{aligned}
n^{n-1}(2^N)^{n-1} &= 2^{nN}\,\frac{n^{n-1}}{2^N} \leq 2^{nN}\,\frac{n^{n-1}}{2^{n\log_2(n+1)-1}}\\
&= 2^{nN}\,\frac{2n^{n-1}}{(n+1)^n} \leq 2^{nN}\,\frac{2}{n+1}\left(\frac{n}{n+1}\right)^{n-1} < 2^{nN}.
\end{aligned}
$$

## The Proof (concluded)

- In summary, $G \in$ HAMILTONIAN CYCLE if and only if $G'$ contains at least $2^{nN}$ cycles.
- $G'$ contains at most $n^n \, 2^{nN}$ cycles.
  - Every $k$-cycle on $G$ induces $(2^N)^k \leq 2^{nN}$ cycles on $G'$.
  - Every cycle on $G'$ is associated with a unique cycle on $G$.
  - There are at most $n^n$ cycles in $G$.
- This number has a polynomial length $O(n^2 \log n)$.
- Hence HAMILTONIAN CYCLE $\in$ P.

# Counting Class #P

A function $f$ is in #P (or $f \in \#\text{P}$) if

- There exists a polynomial-time NTM $M$.
- $M(x)$ has $f(x)$ accepting paths for all inputs $x$.

## Some #P Problems

- $f(\phi)$ = number of satisfying truth assignments to $\phi$.
  - The desired NTM guesses a truth assignment $T$ and accepts $\phi$ if and only if $T \models \phi$.
  - Hence $f \in$ #P.
  - $f$ is also called #SAT.
- #HAMILTONIAN PATH.
- #3-COLORING.

# #P Completeness

- Function $f$ is #P-complete if
  - $f \in \#\text{P}$.
  - $\#\text{P} \subseteq \text{FP}^f$.
    - Every function in #P can be computed in polynomial time with access to a black box[a] for $f$.
      - It said to be polynomial-time Turing-reducible to $f$.
  - Oracle $f$ can be accessed only a polynomial number of times.

---

[a]Think of it as a subroutine. It is also called an **oracle**.

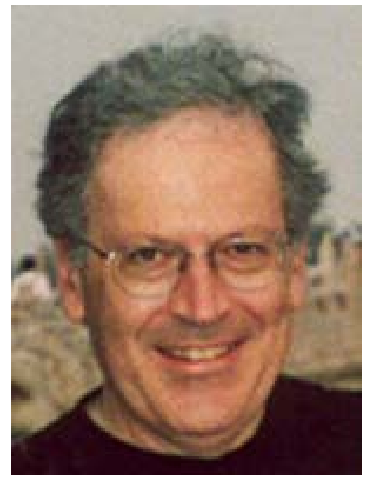
## #SAT Is #P-Complete[a]

- First, it is in #P (p. 859).
- Let $f \in \#\text{P}$ compute the number of accepting paths of $M$.
- Cook's theorem uses a **parsimonious** reduction from $M$ on input $x$ to an instance $\phi$ of SAT.
  - That is, $M(x)$'s number of accepting paths equals $\phi$'s number of satisfying truth assignments.
- Call the oracle #SAT with $\phi$ to obtain the desired answer regarding $f(x)$.

[a]Valiant (1979); in fact, #2SAT is also #P-complete.

## Leslie G. Valiant[a] (1949–)

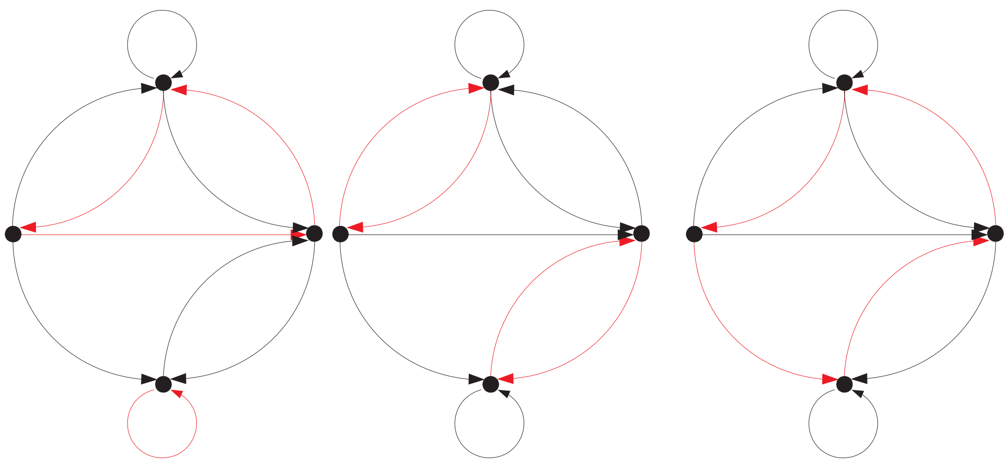
Avi Wigderson (2009), "Les Valiant singlehandedly created, or completely transformed, several fundamental research areas of computer science. [...] We all became addicted to this remarkable throughput, and expect more."



[a]Turing Award (2010).

# 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.

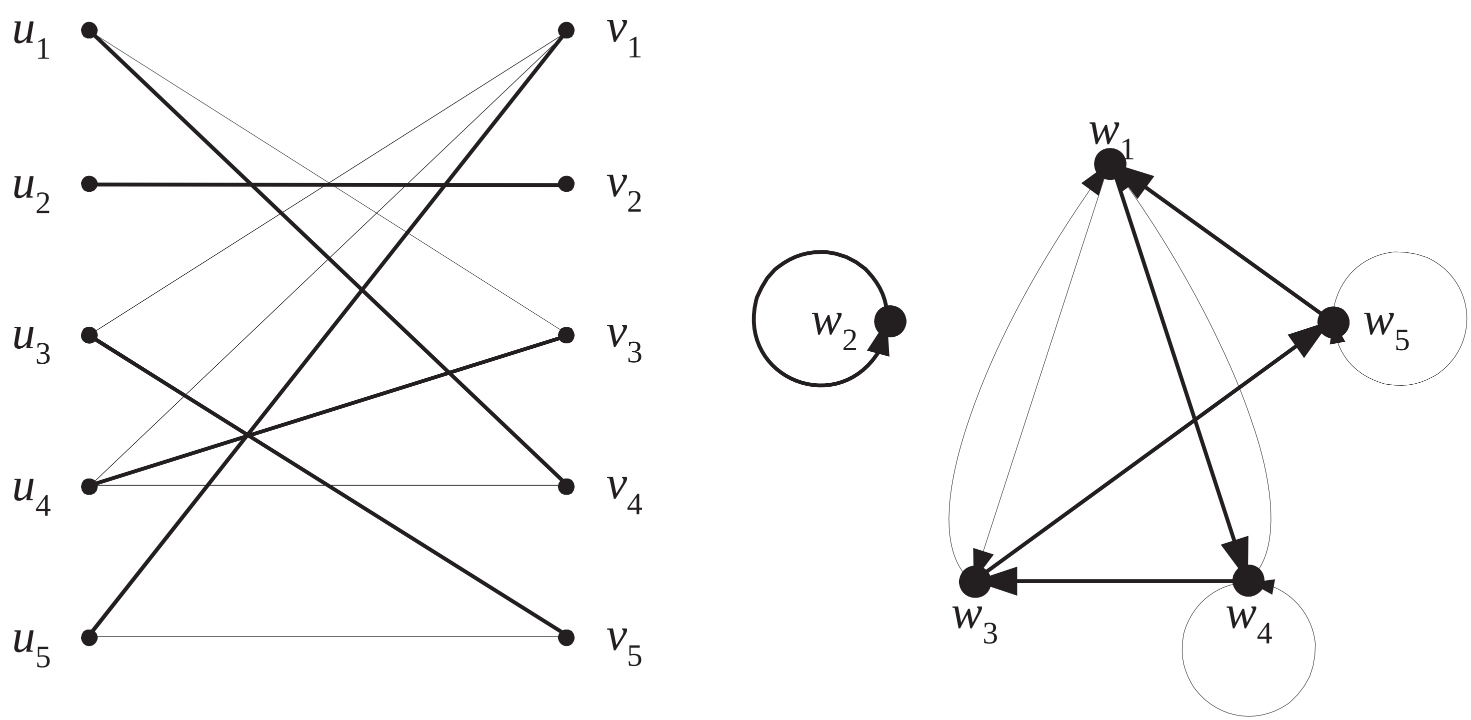
## CYCLE COVER and BIPARTITE PERFECT MATCHING

**Proposition 98** CYCLE COVER *and* BIPARTITE PERFECT MATCHING *(p. 519) are parsimoniously reducible to each other.*

- A polynomial-time algorithm creates a bipartite graph $G'$ from any directed graph $G$.
- Moreover, the number cycle covers for $G$ equals the number of bipartite perfect matchings for $G'$.
- And vice versa.

**Corollary 99** 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. 523: no signs.

- Surprisingly, much harder to compute than determinant!

## Permanent and Counting Perfect Matchings

- BIPARTITE PERFECT MATCHING is related to determinant (p. 524).
- #BIPARTITE PERFECT MATCHING is related to permanent.

**Proposition 100** *0/1* PERMANENT *and* BIPARTITE PERFECT MATCHING *are parsimoniously reducible to each other.*

## 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 $\text{perm}(A) =$ number of perfect matchings in $G$.

# Illustration of the Proof Based on p. 865 (Left)

$$A = \begin{bmatrix} 0 & 0 & 1 & \boxed{1} & 0 \\ 0 & \boxed{1} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & \boxed{1} \\ 1 & 0 & \boxed{1} & 1 & 0 \\ \boxed{1} & 0 & 0 & 0 & 1 \end{bmatrix}.$$

- $\text{perm}(A) = 4$.
- The permutation corresponding to the perfect matching on p. 865 is marked.

## Permanent and Counting Cycle Covers

**Proposition 101** *0/1* PERMANENT *and* CYCLE COVER *are parsimoniously reducible to each other.*

- Let $A$ be the adjacency matrix of the graph on p. 865 (right).
- Then $\text{perm}(A)$ = number of cycle covers.

## Three Parsimoniously Equivalent[a] Problems

We summarize Propositions 98 (p. 864) and 100 (p. 867) in the following.

**Lemma 102** *0/1* PERMANENT, BIPARTITE PERFECT MATCHING, *and* CYCLE COVER *are* **parsimoniously equivalent**.

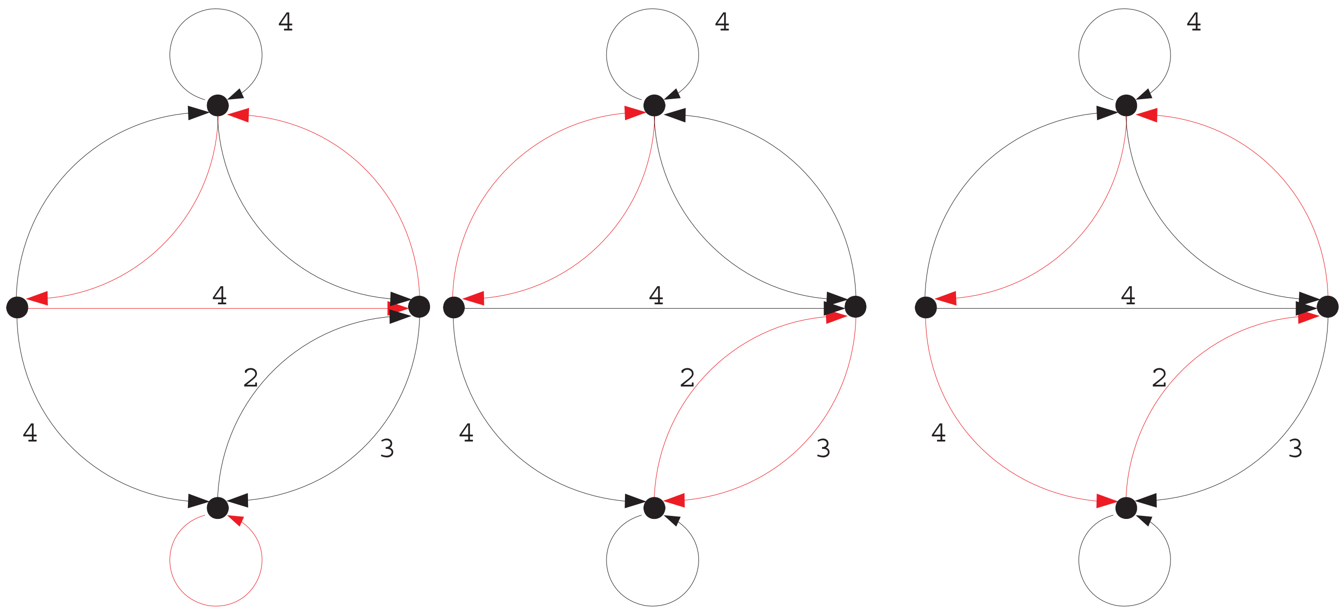
We will show that the counting versions of all three problems are in fact #P-complete.

[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 101 on p. 870).
- #CYCLE COVER is a special case: All weights are 1.

# An Example[a]




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 103 (Valiant, 1979)** 0/1 PERMANENT, #BIPARTITE PERFECT MATCHING, *and* #CYCLE COVER *are #P-complete.*

- By Lemma 102 (p. 871), it suffices to prove that #CYCLE COVER is #P-complete.
- #SAT is #P-complete (p. 861).
- #3SAT is #P-complete because it and #SAT are parsimoniously equivalent.
- We shall prove that #3SAT 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)

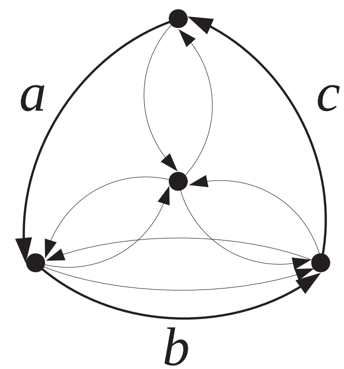
- Our reduction is not expected to be parsimonious.
  - Suppose otherwise and

$$\#\phi = \#G.$$

  - Hence $G$ has a cycle cover if and only if $\phi$ is satisfiable.
  - But CYCLE COVER $\in P$ (p. 864).
  - Thus 3SAT $\in P$, a most unlikely event!

# The Proof: the Clause Gadget (continued)

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




- Each edge has weight 1 unless stated otherwise.
- Each bold edge corresponds to one literal in the clause.
- They are not *parallel* lines as bold edges are schematic only (preview p. 890).

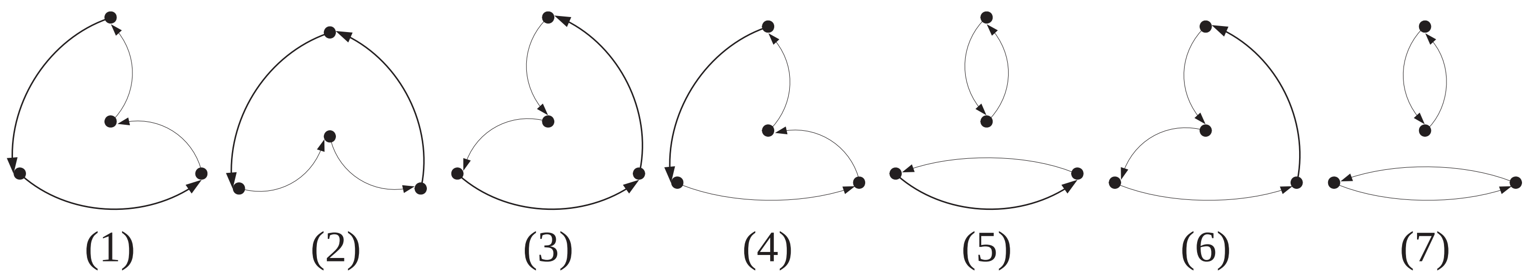
## The Proof: the Clause Gadget (continued)

- Following a bold edge means making 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).

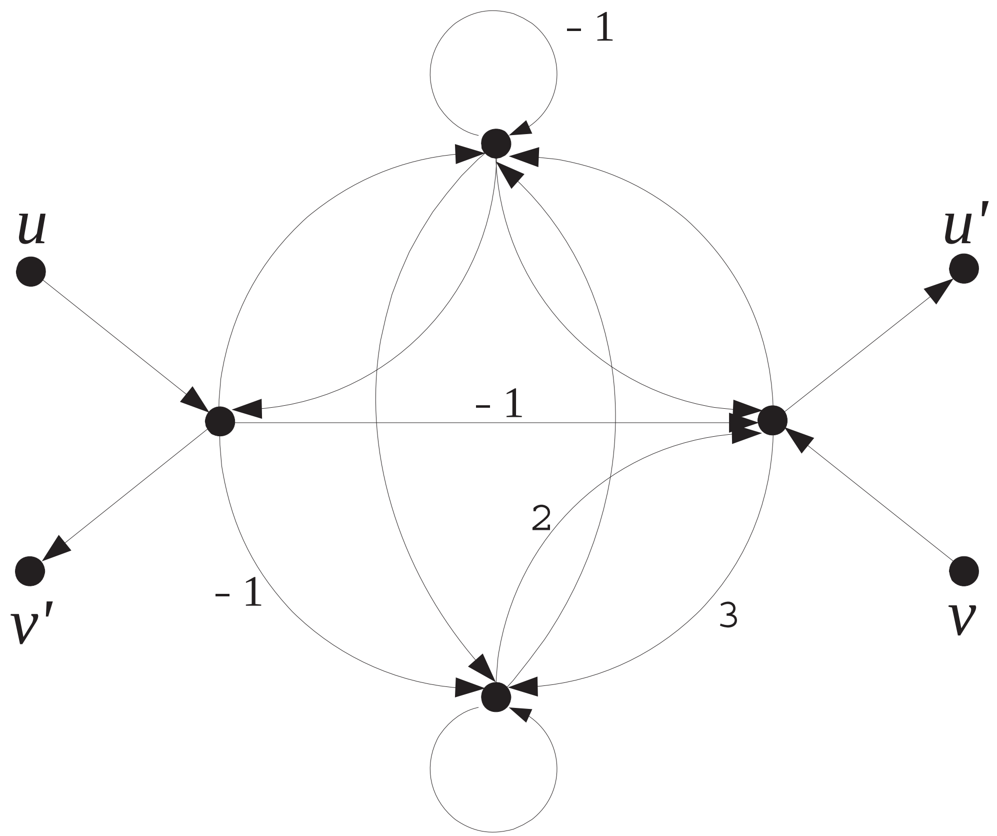
 

## The Proof: the Clause Gadget (continued)

7 possible cycle covers, one for each satisfying assignment: (1) $a = 0, b = 0, c = 1$, (2) $a = 0, b = 1, c = 0$, etc.

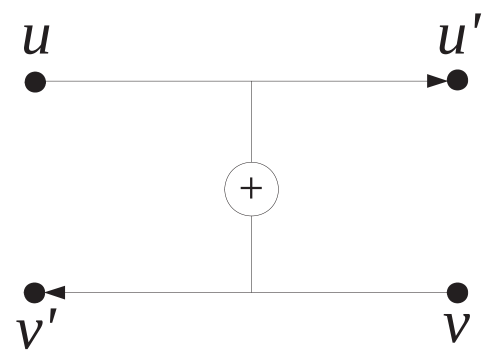


# The Proof: the XOR Gadget (continued)



# The Proof: Properties of the XOR Gadget (continued)

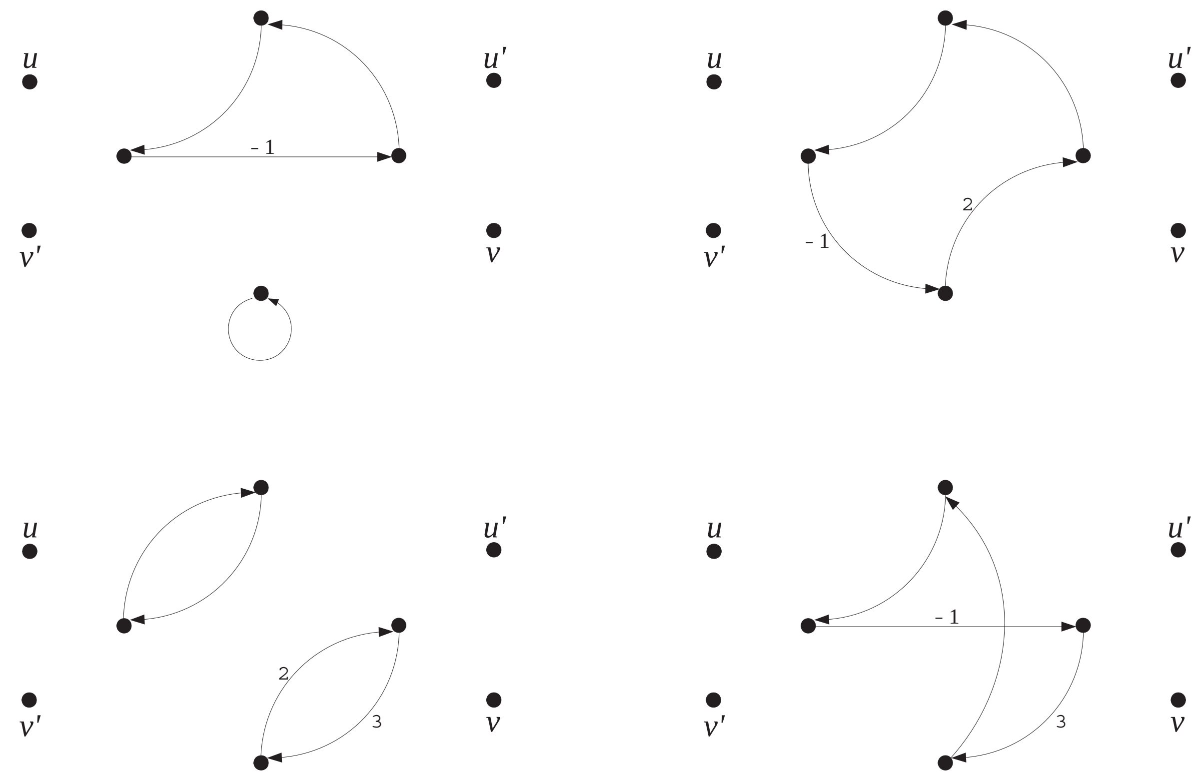
- The XOR gadget schema:




- *At most one* of the 2 schematic edges will be included in a cycle cover.
- Only those cycle covers that take exactly one schematic edge in every XOR gadget will have nonzero weights.
- There will be $3m$ XOR gadgets, one for each literal.

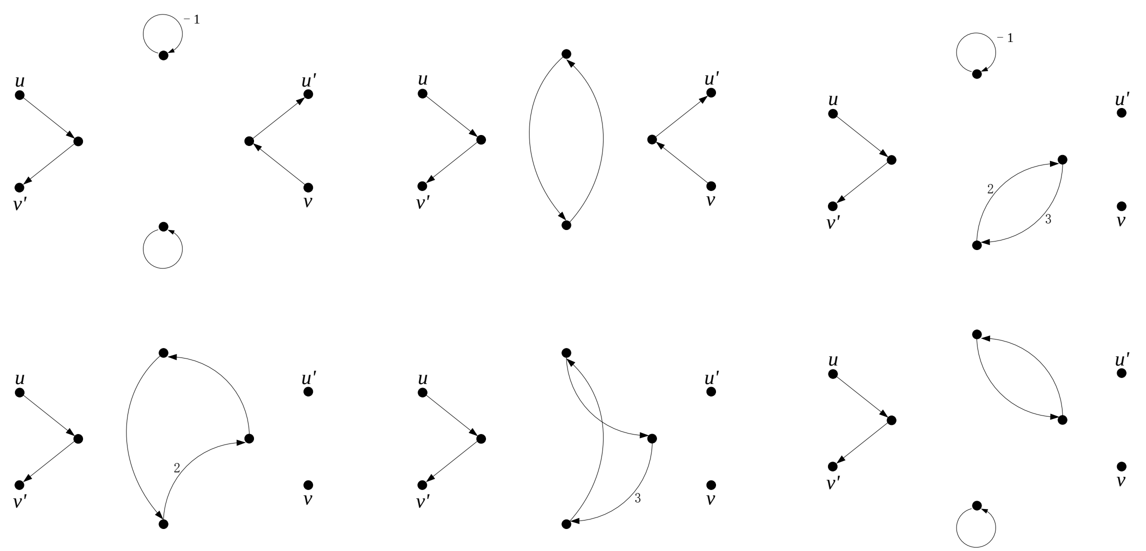
## The Proof: Properties of the XOR Gadget (continued)

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]



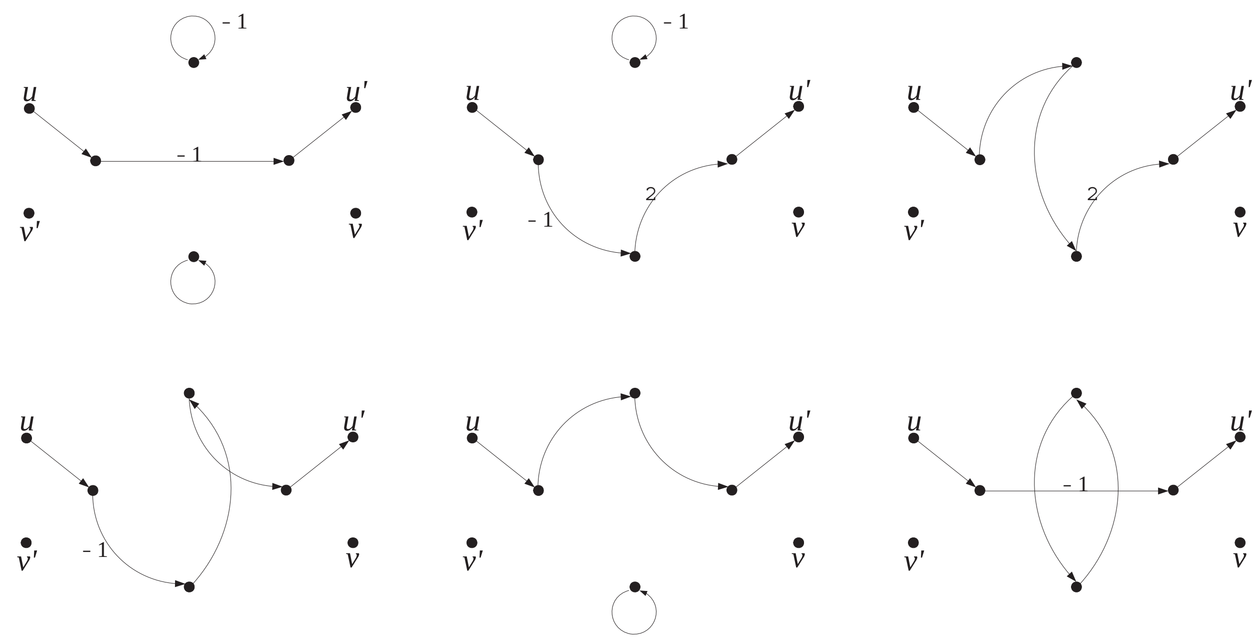

- Same for cycle covers entering at $v$ and leaving at $u'$.

---

[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. 882,

$$\sum_{\text{cycle cover } c \text{ not entering } x} (\text{weight of } c \text{ for } H)$$

$$= \sum_{\text{cycle cover } c \text{ not entering } x} (\text{weight of } c \text{ for } H - x) \times (\text{weight of } c \text{ for } x)$$

$$= \sum_{\text{cycle cover } c \text{ not entering } x} (\text{weight of } c \text{ for } H - x) \times 0 = 0.$$

## The Proof: Summary (continued)

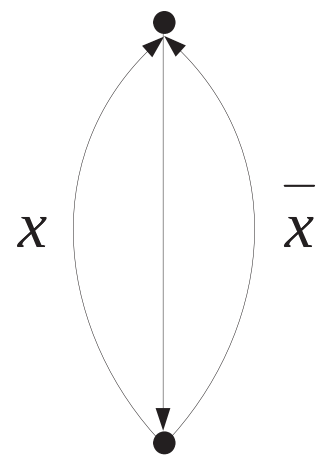
- Cycle covers entering *any* of the XOR gadgets and leaving illegally contribute 0 to the cycle count by p. 883.
- For every XOR gadget entered and exited legally, the total weight of a cycle cover is multiplied by 4.
  - Each such act multiplies the weight by 4 according to p. 884.

## 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.

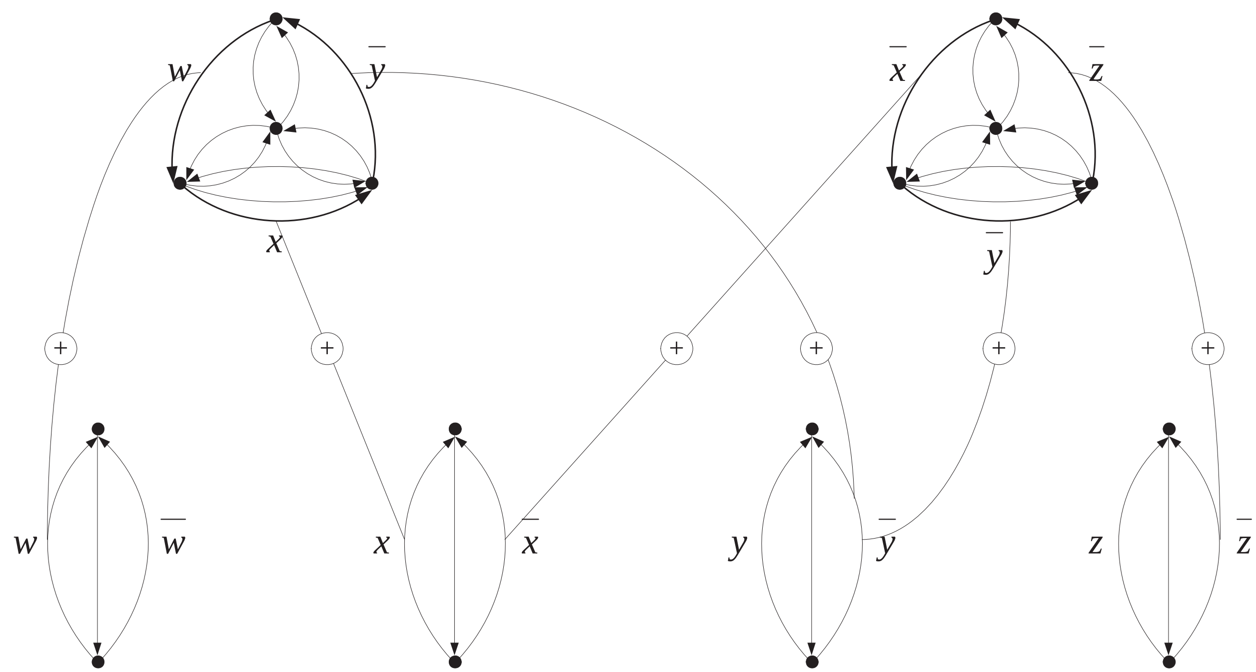
# The Proof: the Choice Gadget (continued)

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

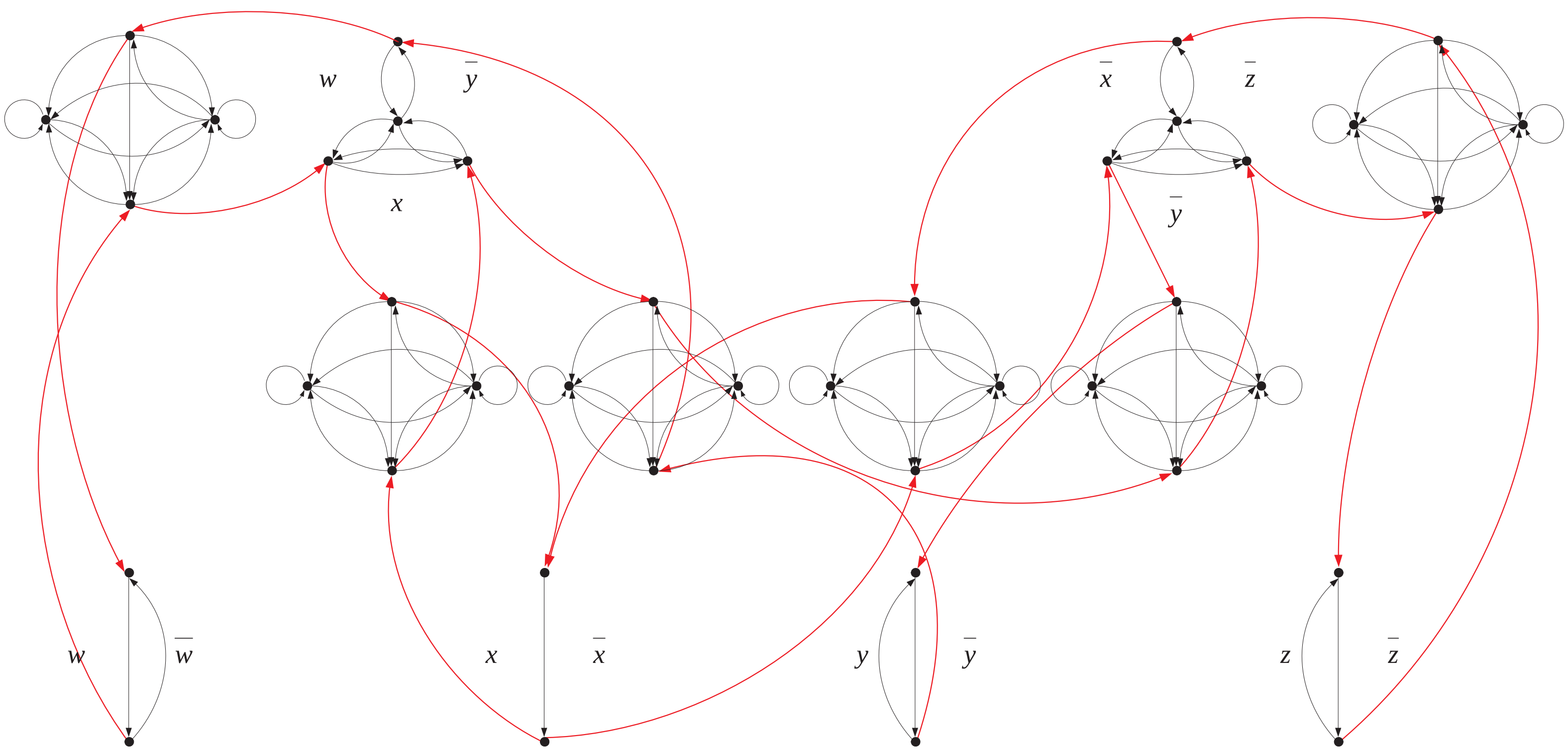



- It gives the truth assignment for the variable.
- Use it with the XOR gadget to enforce consistency.

## Schema for $(w \lor x \lor \bar{y}) \land (\bar{x} \lor \bar{y} \lor \bar{z})$



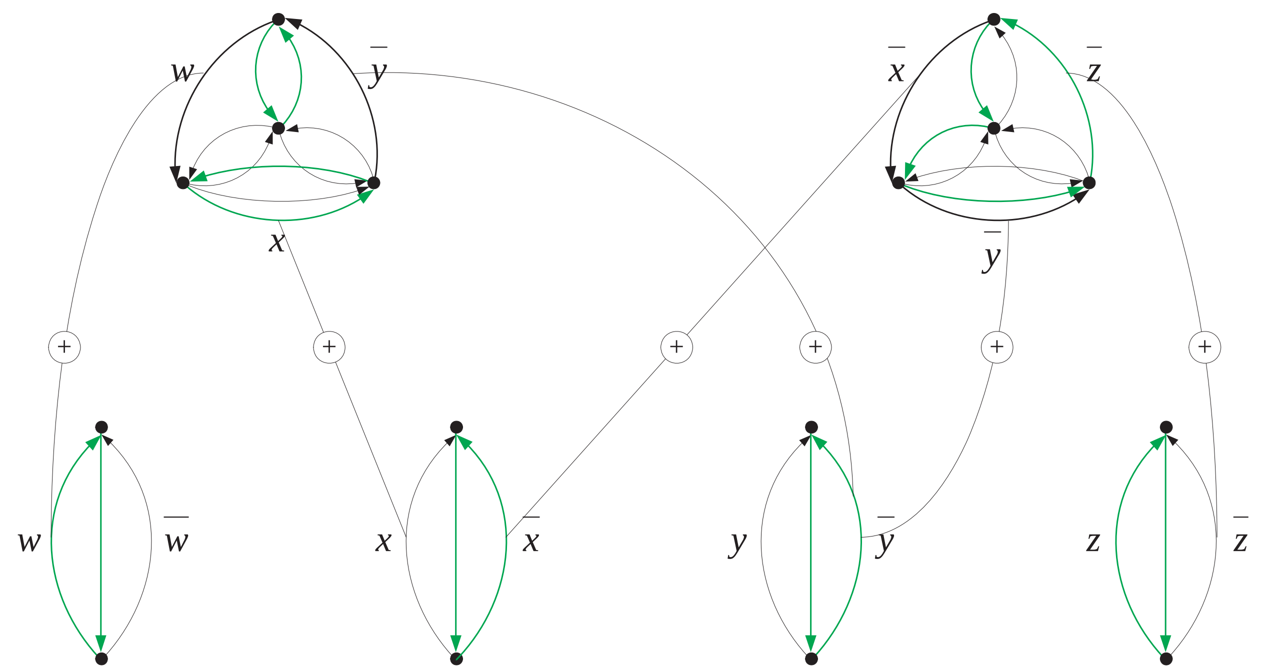
# Full Graph $(w \vee x \vee \bar{y}) \wedge (\bar{x} \vee \bar{y} \vee \bar{z})$



## The Proof: a Key Observation (continued)

Each satisfying truth assignment to $\phi$ corresponds to a schematic cycle cover that respects the XOR gadgets.

## $w = 1, x = 0, y = 0, z = 1 \Leftrightarrow$ One Cycle Cover

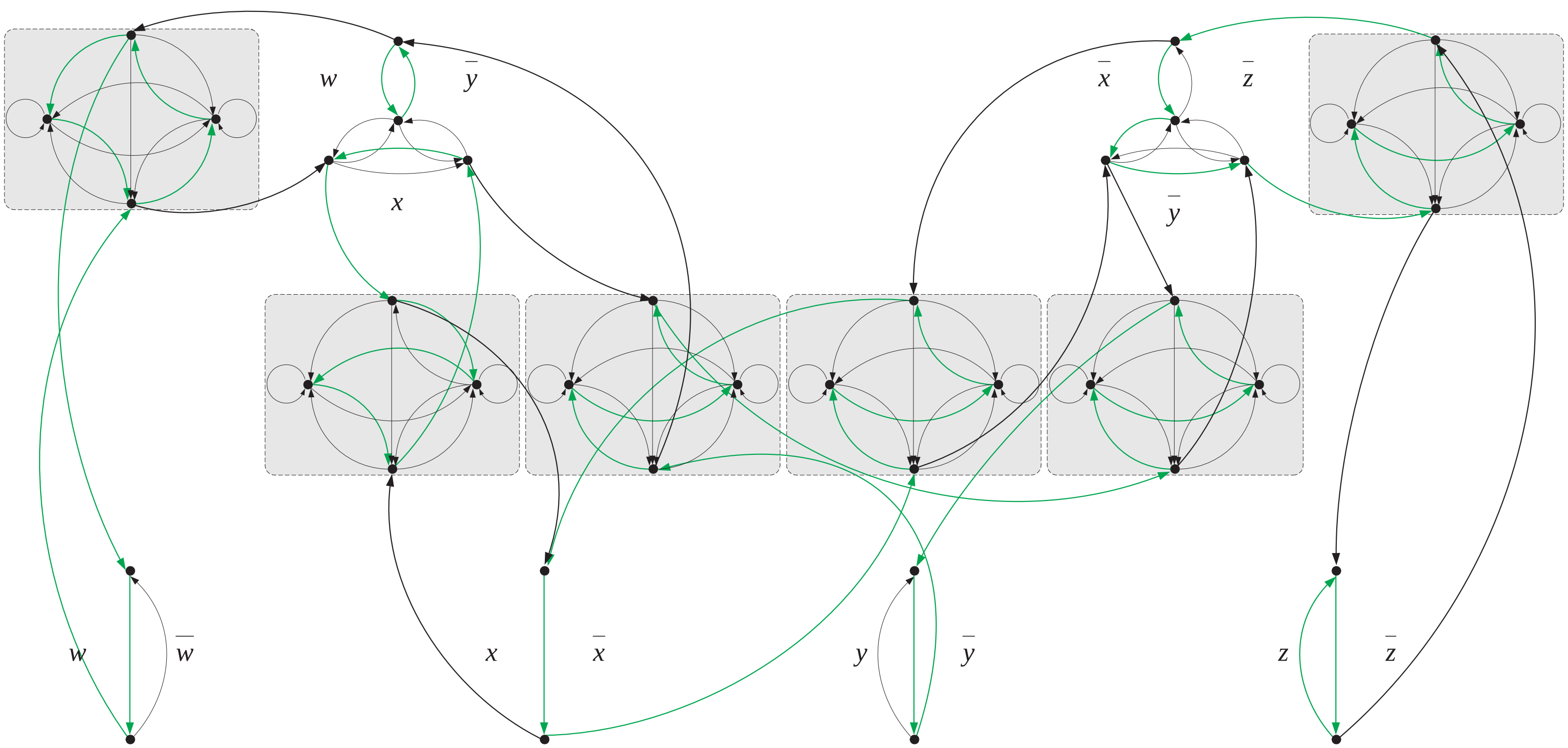


## The Proof: a Key Corollary (continued)

- Recall that there are $3m$ XOR gadgets.
- Each satisfying truth assignment to $\phi$ contributes $4^{3m}$ to the cycle count $\#H$.
- Hence
$$\#H = 4^{3m} \times \#\phi, \tag{27}$$
as desired.

# "$w = 1, x = 0, y = 0, z = 1$" Adds $4^6$ to Cycle Count



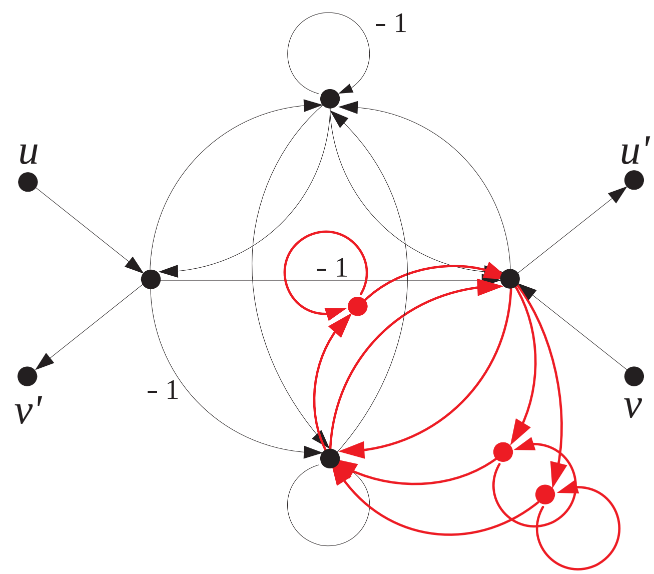
## The Proof (continued)

- We are almost done.
- The weighted directed graph $H$ needs to be *efficiently* replaced by some unweighted graph $G$.
- Furthermore, knowing $\#G$ should enable us to calculate $\#H$ *efficiently*.
  - This done, $\#\phi$ will have been Turing-reducible to $\#G$.[a]
- We proceed to construct this graph $G$.

[a]By way of $\#H$.

# The Proof: Construction of $G$ (continued)

- Replace edges with weights 2 and 3 as follows (note that the graph cannot have parallel edges):




- 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 contains 4 nodes (p. 877), and there are $m$ of them (one per clause).
- Each revised XOR gadget contains 7 nodes (p. 896), and there are $3m$ of them (one per literal).
- Each choice gadget contains 2 nodes (p. 888), and there are $n \leq 3m$ of them (one per variable).
- So

$$M \leq 4m + 21m + 6m = 31m.$$

## The Proof: Construction of $G$ (continued)

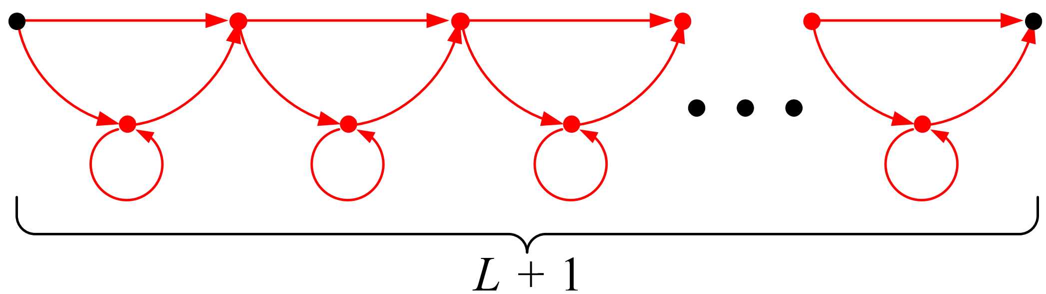
- $\#H \leq 2^L$ for some $L = O(m \log m)$.
  - The maximum absolute value of the edge weight is 1.
  - Hence each term in the permanent is at most 1.
  - There are $M! \leq (31m)!$ terms.
  - Hence

$$\begin{aligned} \#H &\leq \sqrt{2\pi(31m)} \left(\frac{31m}{e}\right)^{31m} e^{\frac{1}{12\times(31m)}} \\ &= 2^{O(m \log m)} \end{aligned} \tag{28}$$

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 $2^{L+1}$:

$$\#H = \cdots + \overbrace{1 \cdot 1 \cdots (-1) \cdots 1}^{\text{a cycle cover}} + \cdots,$$

$$\#G = \cdots + \overbrace{1 \cdot 1 \cdots 2^{L+1} \cdots 1}^{\text{a cycle cover}} + \cdots.$$

- 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. 898).
- 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. (27) on p. 893.
- So

$$\#\phi = \frac{a_0 - a_1 + a_2 - \cdots + (-1)^n a_n}{4^{3m}}.$$

  – Equivalently,

$$\#\phi = \frac{\#G \bmod (2^{L+1} + 1)}{4^{3m}}.$$

*Finis*