

## Lecture 8 : Zero-one permanent is #P-complete

Lecturer: Jayalal Sarma M.N.

Scribe: T Devanathan

NOTE : DRAFT VERSION - UNEDITED

## 1 Permanent of (0,1) matrix is #P-complete

In this lecture, we will discuss the proof given by Valiant for the #P-completeness of permanent. Recall the following the results we proved in previous lectures.

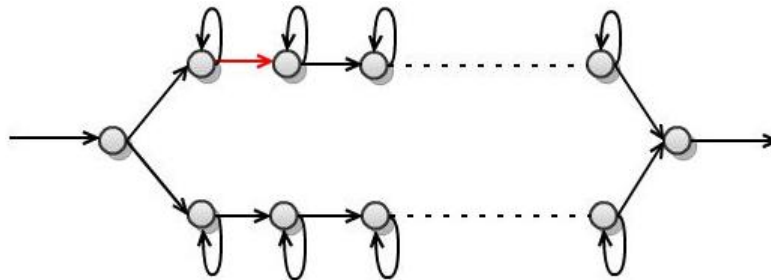
**Lemma 1.** For  $A \in \{0,1\}^{n \times n}$ ,  $\text{perm}(A) \in \#P$ .

**Lemma 2.** Let  $A \in \mathbb{Z}^{n \times n}$ . Let  $G$  be the directed weighted graph on  $n$  vertices obtained by interpreting  $A$  as the weighted adjacency matrix of  $G$ , i.e.  $\forall 1 \leq i, j \leq n, wt(i, j) = w$  if and only if  $A_{i,j} = w$ . Let  $\mathcal{C}$  be the set of cycle covers of  $G$ . Then,  $\text{perm}(A) = \sum_{C \in \mathcal{C}} wt(C)$ .

**Theorem 3.** Computing the permanent of a (0,1)-matrix is #P-complete.

*Proof.* We will describe a reduction from #SAT to #CC( $G$ ). Given a 3-CNF formula  $\phi = C_1 \wedge C_2 \wedge \dots \wedge C_n$ , we will construct a matrix  $A_\phi$  such that number of satisfying assignments of  $\phi$  is equal to #CC( $G$ ) where  $G$  is the graph with  $A_\phi$  as its adjacency matrix.  $G$  is constructed using 3 different kinds of gadgets.

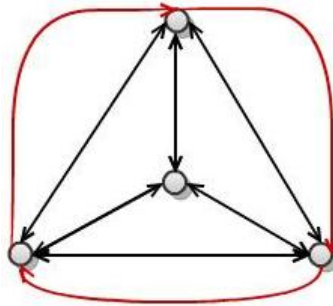
1. There is a variable gadget corresponding to each variable in  $\phi$ .



Observe that a cycle cover for the vertices in this gadget should contain exactly one of the two paths. For the vertices in the other path, the self loops are to be considered in the cover. Note that no cycle cover can choose self loops in both paths and cover the entry and exit nodes. Without loss of generality, assume that the path above

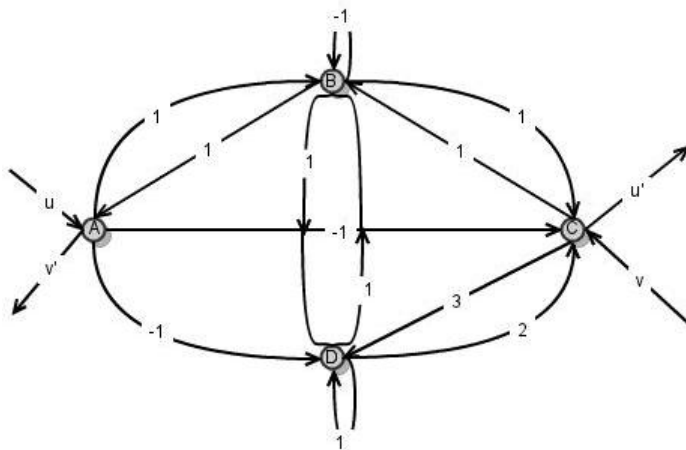
corresponds to the variable being 1 and the one below corresponds to the variable being 0. Connect all the variables in the expression in a loop.

2. There is a clause gadget corresponding to each clause in the expression.



Observe that, to cover all four vertices, at least one of the three outer edges coloured in red must be left out. Associate each outer edge to a variable in the clause and assume that the edge is not chosen in the cover if the corresponding variable is 1. This ensures that every clause has at least one variable that is 1 and hence the existence of cycle cover implies the existence of a satisfying assignment to the expression.

3. To ensure that the value of the variable is consistent between the variable and clause gadgets, introduce a connector gadget as shown below.



In the clause gadget, the outer edge corresponding to a variable is removed and the above gadget is introduced so that the vertices adjacent to the removed edge in the clause gadget are connected at  $u$  and  $u'$ . Similarly, in the variable gadget, one edge from the top path, say the edge coloured in red, is removed and this gadget substituted in its place so that the vertices adjacent to the removed edge are connected at  $v$  and  $v'$ . The construction of the connector is such that both  $u$  to  $u'$  and  $v$  to  $v'$  would not

be in the cycle cover simultaneously. This ensures the consistency of a variable value between the two gadgets.

To show that  $\#CC(G) = \#\text{SAT}(\phi)$ , we have to verify if the connector gadget allows only paths  $u$  to  $u'$  or  $v$  to  $v'$  to contribute to the cycle cover and that the connector gadget doesn't contribute to the cycle cover if neither of the paths is chosen. Consider the following cases and the corresponding ways for vertices to be a part of cycle cover and the corresponding contribution to the final sum:

1. None of  $u$ ,  $u'$ ,  $v$  or  $v'$  is used.

- (a) (AB)(DC) contributes 6
- (b) (ADCB) contributes -2
- (c) (ACB)(D) contributes -1
- (d) (ACDB) contributes -3

Net contribution = 0.

2. Cover passes from  $u$  to  $v'$ .  $v$  and  $u'$  are not used.

- (a) (BC)(D) contributes 1
- (b) (CD)(B) contributes -6
- (c) (BCD) contributes 3
- (d) (BDC) contributes 2

Net contribution = 0.

3. Cover passes from  $v$  to  $u'$ .  $v'$  and  $u$  are not used.

- (a) (AB)(D) contributes 1
- (b) (ADB) contributes -1

Net contribution = 0.

4. Cover passes from  $u$  to  $v'$  and also  $v$  to  $u'$ .

- (a) (BD) contributes 1
- (b) (D)(B) contributes -1

Net contribution = 0.

5. Cover passes from  $v$  to  $v'$  and hence  $u$  and  $u'$  not used.

- (a) (CDBA) contributes 3

(b) (CBA) contributes 1

Net contribution = 4.

6. Cover passes from  $u$  to  $u'$  and hence  $v$  and  $v'$  not used.

(a) (ABC)(D) contributes 1

(b) (AC)(BD) contributes -1

(c) (AC)(B)(D) contributes 1

(d) (ADC)(B) contributes 2

(e) (ADBC) contributes -1

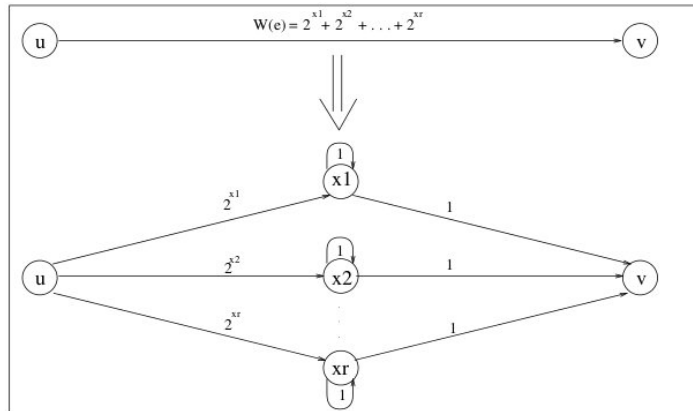
(f) (ABDC) contributes 2

Net contribution = 4.

Observe that, for every variable in the clause, there is a contribution of 4 to the solution of  $\#CC$ . Therefore,  $\#CC(G) = 4^{3m}\#\text{SAT}(\phi)$  as there are  $3m$  connector gadgets in total in the graph ( $m$  clauses in the 3-CNF). Thus,  $\#\text{SAT}$  is reduced to  $\#CC$  which is equal to  $\text{perm}(A)$ . However, in this argument, the matrix is over  $\{-1, 0, 1, 2, 3\}$ . The problem of computing permanent of an integer matrix can be reduced to the problem of permanent computation of  $(0, 1)$ -matrix as follows.

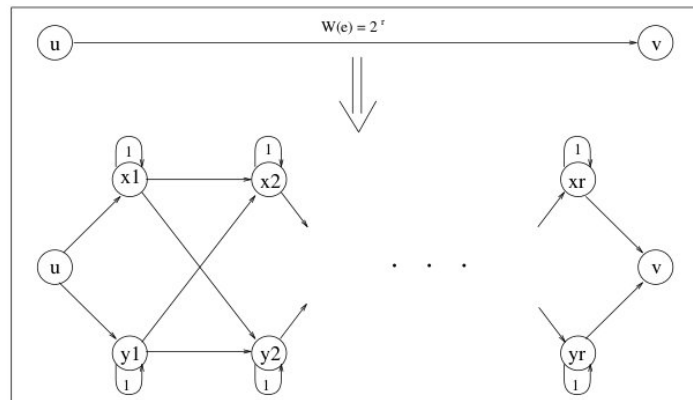
*Remark 4.* We remark that the above reduction gives the following observation : for a given graph is the sum of weights of cycle covers is positive is as hard as SAT.

1. **Reduction from integer matrix to non-negative integer matrix:** Let  $A$  be an  $n \times n$  integer matrix in which no entry is larger than  $\mu$  in magnitude. From the definition of the permanent, it follows that  $|\text{perm}(A)| \leq n!\mu^n$ . To compute  $\text{perm}(A)$  it is sufficient to compute its value  $\pmod{Q}$  for  $Q > 2n!\mu^n$ . Formally, given  $A$  compute  $Q = 2n!\mu^n + 1$ ,  $A' = A \pmod{Q}$  and  $P = \text{perm}(A') \pmod{Q}$ . If  $P < Q/2$  then  $\text{perm}(A) = P$ . Otherwise  $\text{perm}(A) = P - Q$  (value is negative). This transformation is done in polynomial time.
2. **Reduction from non-negative integer matrix to matrix with only non-negative powers of two:** Replace edges as follows: if the weight of the edge is  $w$ , such that,  $w = 2^{x_1} + 2^{x_2} + \dots + 2^{x_r}$ , replace the edge by the following construct.



This doesn't affect the value of the sum of cycle covers.

3. **Reduction from matrix with only non-negative powers of two matrix to a (0, 1)- matrix:** Replace edges as follows: if the weight of the edge is  $2^m$ , replace it with the following construct.



Once again, observe that the value of the sum of cycle covers doesn't change and the resultant graph  $\in \{0, 1\}^{n \times n}$ .

Thus, for an integer matrix  $A$ , there exists an  $(0, 1)$ -matrix  $B$ , such that,  $\text{perm}(B) = \text{perm}(A) \pmod{Q}$ . Hence, from **Lemma 1**, we can conclude that  $\text{perm}(A)$  is #P-complete, if  $A$  is a  $(0, 1)$ -matrix.  $\square$

*Remark 5.* Why does this not contradict the fact that we can efficiently test if  $\text{perm}$  is positive for 0-1 matrices. Notice that the  $\pmod{Q}$  does the trick. The value of the permanent will be computed only  $\pmod{Q}$ . The actual value of the permanent after substitution could be much larger, and we are guaranteed equality only  $\pmod{Q}$ . Hence a zero value under this computation does not mean that permanent value of the matrix is zero, it just means that it is a multiple of  $Q$ .