Every $\#\text{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

An instance of $\#\text{CSP}_q(\mathcal{F})$ has the following bipartite view.

- Create a node for each variable and each constraint.

Every $\#\text{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

An instance of $\#\text{CSP}_q(\mathcal{F})$ has the following bipartite view.

- Create a node for each variable and each constraint.
- Connect a variable node to a constraint node if the variable appears in the constraint.

Every $\#\text{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

An instance of $\#\text{CSP}_q(\mathcal{F})$ has the following bipartite view.

- Create a node for each variable and each constraint.
- Connect a variable node to a constraint node if the variable appears in the constraint.
- This graph is called the constraint graph.

Every $\#\mathrm{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

An instance of $\#\mathrm{CSP}_q(\mathcal{F})$ has the following bipartite view.

- Create a node for each variable and each constraint.
- Connect a variable node to a constraint node if the variable appears in the constraint.
- This graph is called the constraint graph.
- To each variable we assign the *Equality* signature of the appropriate arity (which is equal to the number of occurrences of the variable in all constraints).

Every $\#\mathrm{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

An instance of $\#\mathrm{CSP}_q(\mathcal{F})$ has the following bipartite view.

- Create a node for each variable and each constraint.
- Connect a variable node to a constraint node if the variable appears in the constraint.
- This graph is called the constraint graph.
- To each variable we assign the *Equality* signature of the appropriate arity (which is equal to the number of occurrences of the variable in all constraints).
- To each constraint node we assign the constraint function in $\mathcal{F}$ used in that constraint.

Every $\#\mathrm{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

An instance of $\#\mathrm{CSP}_q(\mathcal{F})$ has the following bipartite view.

- Create a node for each variable and each constraint.
- Connect a variable node to a constraint node if the variable appears in the constraint.
- This graph is called the constraint graph.
- To each variable we assign the *Equality* signature of the appropriate arity (which is equal to the number of occurrences of the variable in all constraints).
- To each constraint node we assign the constraint function in $\mathcal{F}$ used in that constraint.

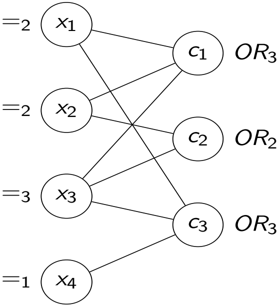
$\#\mathrm{CSP}_q(\mathcal{F}) \leq_T \mathrm{Holant}_q(\mathcal{EQ}_k \mid \mathcal{F})$, where $\mathcal{EQ}_k = \{=_k \mid k \geq 1\}$.

Every $\#\mathrm{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

For example, $(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee x_4)$ which corresponds to

1. variables x_1, x_2, x_3, x_4,
2. constraints $(OR_3, x_1, x_2, x_3), (OR_2, x_2, x_3), (OR_3, x_1, x_3, x_4)$,

becomes

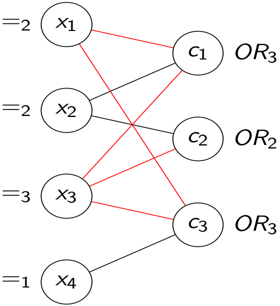


Every $\#\mathrm{CSP}_q(\mathcal{F})$ can be expressed as a Holant problem

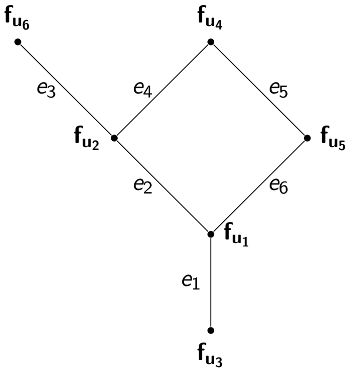
For example, $(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee x_4)$ which corresponds to

1. variables x_1, x_2, x_3, x_4,
2. constraints $(OR_3, x_1, x_2, x_3), (OR_2, x_2, x_3), (OR_3, x_1, x_3, x_4)$,

becomes



Every Holant problem can be expressed as a #CSP



$\Longrightarrow$

- Variables $e_1, e_2, \ldots, e_6$
- Constraints
  - (f_{u_1}, e_1, e_2, e_6)
  - (f_{u_2}, e_2, e_3, e_4)
  - (f_{u_3}, e_1)
  - (f_{u_4}, e_4, e_5)
  - (f_{u_5}, e_5, e_6)
  - (f_{u_6}, e_3)

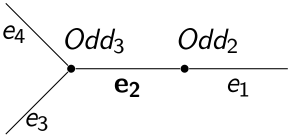
$\text{Holant}_q(\mathcal{F}) \leq_T \#\text{CSP}_q(\mathcal{F})$, where every variable appears exactly in 2 constraints.

Overview

Local gadget constructions via $\mathcal{F}$-gates

In what follows we focus on functions over the Boolean domain that take values in $\mathbb{C}$.

- In the context of Holant problems, we construct a graph in order to realize a signature.
- We say a signature f is realizable from $\mathcal{F}$ if there is a graph with some dangling edges, where each vertex is assigned a signature from $\mathcal{F}$, and the resulting signature with inputs on the dangling edges is exactly f.
- For example, $Even_3$ is realizable from $\mathcal{F} = \{Odd_3, Odd_2\}$.



- We call this an $\mathcal{F}$-gate.

$\mathcal{F}$-gates

An $\mathcal{F}$-gate F is similar to a signature grid (G, π) except that $G = (V, E, E')$ with regular edges E and some dangling edges E', and $E \cap E' = \emptyset$.

Each dangling edge $e' \in E'$ has only one end incident to a vertex in V, and the other edge dangling.

$\mathcal{F}$-gates

An $\mathcal{F}$-gate F is similar to a signature grid (G, π) except that $G = (V, E, E')$ with regular edges E and some dangling edges E', and $E \cap E' = \emptyset$.

Each dangling edge $e' \in E'$ has only one end incident to a vertex in V, and the other edge dangling.

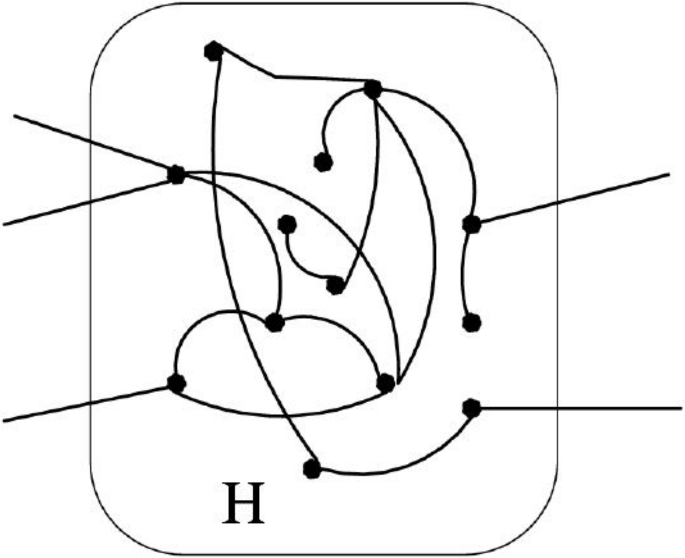
An $\mathcal{F}$-gate F with k dangling edges defines the function

$$\Gamma(y_1, \ldots, y_k) = \sum_{\sigma: E \to [2]} \prod_{u \in V} f_u(\hat{\sigma} \upharpoonright_{E(u)})$$

where $(y_1, \ldots, y_k) \in [2]^k$ is an assignment on the dangling edges, $\hat{\sigma}$ is the extension of the assignment σ on the internal edges E by the assignment $(y_1, \ldots, y_k)$ on the dangling edges E'.

We call this function Γ the signature of this $\mathcal{F}$-gate.

An example of an $\mathcal{F}$-gate with underlying graph H



The signature matrix

Let f be a signature of arity 2 over the Boolean domain. Then f can be written as a vector $f = (f_{00}, f_{01}, f_{10}, f_{11})$ or as a matrix $M = \begin{bmatrix} f_{00} & f_{01} \\ f_{10} & f_{11} \end{bmatrix}$.

For example, if F, F' are the following $\mathcal{F}$-gates

$$\underset{e_1}{\overline{\qquad\quad}}\; F \;\underset{e_2}{\overline{\qquad\quad}} \quad \text{with } M = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix}$$

$$\underset{e_1'}{\overline{\qquad\quad}}\; F' \;\underset{e_2'}{\overline{\qquad\quad}} \quad \text{with } M' = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

then $F(0,0) = 0$, $F(0,1) = 1$, $F(1,0) = 1$ and $F(1,1) = 3$,
$F'(0,0) = 2$, $F'(0,1) = 0$, $F'(1,0) = 1$ and $F'(1,1) = 1$.

Multiplication of signature matrices

$$\underset{e_1}{\overline{\qquad}}\ F\ \underset{e_2}{\overline{\qquad}} \quad \text{with } M = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix}$$

$$\underset{e_1'}{\overline{\qquad}}\ F'\ \underset{e_2'}{\overline{\qquad}} \quad \text{with } M' = \begin{bmatrix} 2 & 0 \\ 1 & 1 \end{bmatrix}$$

The matrix product $MM' = \begin{bmatrix} 1 & 1 \\ 5 & 3 \end{bmatrix}$ is the signature matrix of the $\mathcal{F}$-gate

$$\underset{e_1}{\overline{\qquad}}\ F\ \underset{\mathbf{e}}{\overline{\qquad}}\ F'\ \underset{e_2'}{\overline{\qquad}}$$

which links F and F' by merging e_2 and e_1'.

Multiplication of signature matrices

$$MM' = \begin{bmatrix} M_{00}M'_{00} + M_{01}M'_{10} & M_{00}M'_{01} + M_{01}M'_{11} \\ M_{10}M'_{00} + M_{11}M'_{10} & M_{10}M'_{01} + M_{11}M'_{11} \end{bmatrix}$$

$\underset{e_1}{\text{———}}\; F \;\underset{e_2}{\text{———}}$ $\qquad\qquad$ $\underset{e'_1}{\text{———}}\; F' \;\underset{e'_2}{\text{———}}$

$$\Downarrow$$

$\underset{e_1}{\text{———}}\; F \;\underset{\mathbf{e}}{\text{———}}\; F' \;\underset{e'_2}{\text{———}}$

What about gates that have a different number of left and right dangling edges?

- A gate with l left dangling edges and $n - l$ right dangling edges can be represented by a $2^l \times 2^{n-l}$ matrix.

- More specifically,
  - a gate with only n left dangling edges can be represented as a $2^n \times 1$ matrix and it is called contravariant or a generator,

  - a gate with only n right dangling edges can be represented as a 1×2^n matrix and it is called covariant or a recognizer.

Kronecker product of matrices

- Let A be the 3×3 matrix $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$,

- and B be the 2×2 matrix $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$.

Kronecker product of matrices

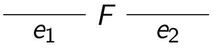
- Then the Kronecker product of A and B is the following 6×6 matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & a_{13}B \\ a_{21}B & a_{22}B & a_{23}B \\ a_{31}B & a_{32}B & a_{33}B \end{bmatrix} =$$

.

$$\begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} & a_{13}b_{11} & a_{13}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} & a_{13}b_{21} & a_{13}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} & a_{23}b_{11} & a_{23}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} & a_{23}b_{21} & a_{23}b_{22} \\ a_{31}b_{11} & a_{31}b_{12} & a_{32}b_{11} & a_{32}b_{12} & a_{33}b_{11} & a_{33}b_{12} \\ a_{31}b_{21} & a_{31}b_{22} & a_{32}b_{21} & a_{32}b_{22} & a_{33}b_{21} & a_{33}b_{22} \end{bmatrix}.$$

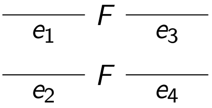
Kronecker product of a signature matrix

- Let F be the $\mathcal{F}$-gate 

with $M = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix}$.

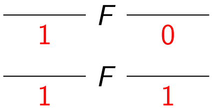
- Then $M^{\otimes 2} = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 1 & 0 & 3 \\ 1 & 3 & 3 & 9 \end{pmatrix} \end{array} \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array}$

  is the signature matrix of the $\mathcal{F}$-gate



Kronecker product of a signature matrix

For example, the following assignment on the edges:



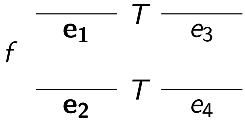
corresponds to the content of the $(11, 01)$ cell of the matrix $M^{\otimes 2}$:

$$M^{\otimes 2} = \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 3 \end{bmatrix} = \begin{array}{c} \begin{matrix} 00 & \mathbf{01} & 10 & 11 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 3 \\ 0 & 1 & 0 & 3 \\ 1 & 3 & 3 & 9 \end{pmatrix} \end{array} \begin{matrix} 00 \\ 01 \\ 10 \\ \mathbf{11} \end{matrix}$$

which is equal to the product $M_{10}M_{11} = 1 \cdot 3 = 3$.

Operations with signature matrices

- Let the row vector $(0, 1, 1, 0)$ represent the signature f of a gate with two right dangling edges.
- Let $T = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$ be the matrix representing the signature of a gate with one left dangling edge and one right dangling edge.
- Then the signature of the following gate with two right dangling edges



is given by the matrix $f \cdot T^{\otimes 2} = (0, 1, 1, 2)$.

Holographic transformations

- We consider bipartite graphs. We transform a general graph into a bipartite graph while preserving the Holant value:
  - For each edge in the graph, we replace it by a path of length two (2-stretch of the graph, edge-vertex incidence graph).
  - Each new vertex is assigned the binary *Equality* signature $(=_2)$.

Holographic transformations

- We consider bipartite graphs. We transform a general graph into a bipartite graph while preserving the Holant value:
  - For each edge in the graph, we replace it by a path of length two (2-stretch of the graph, edge-vertex incidence graph).
  - Each new vertex is assigned the binary *Equality* signature ($=_2$).
- Let T be an invertible 2×2 matrix. The holographic transformation defined by T is the following operation:
  - Given $\Omega = (H, \pi)$ of Holant($\mathcal{F} \mid \mathcal{G}$), we get a new signature grid $\Omega' = (H, \pi')$ of Holant($\mathcal{F}T \mid T^{-1}\mathcal{G}$) by replacing each signature $f_u \in \mathcal{F}$ (resp. $g_v \in \mathcal{G}$) by $f_u \cdot T^{\otimes deg(u)}$ (resp. $(T^{-1})^{\otimes deg(v)} \cdot g_v$).

Valiant's Holant Theorem

Theorem

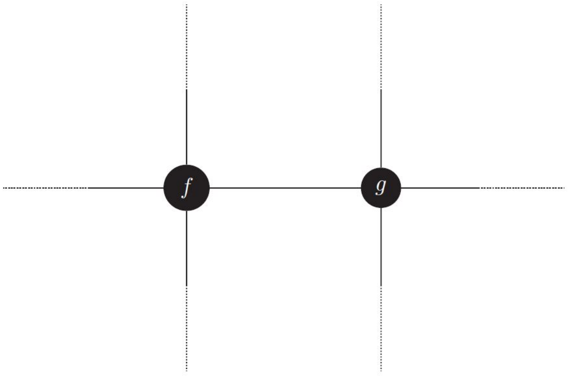
Let $\mathcal{F}$ and $\mathcal{G}$ be sets of complex-valued signatures over a Boolean domain. Suppose Ω is a bipartite signature over $(\mathcal{F} \mid \mathcal{G})$. If T is an invertible 2×2 matrix over $\mathbb{C}$, then

$$\mathit{Holant}(\Omega, \mathcal{F} \mid \mathcal{G}) = \mathit{Holant}(\Omega', \mathcal{F}T \mid T^{-1}\mathcal{G})$$

where Ω' is the corresponding signature grid over $(\mathcal{F}T \mid T^{-1}\mathcal{G})$.

Proof.

1. Let $\Omega_0 = \Omega$. Let $G = (U, V, E)$ be the underlying graph. An edge $e = (u, v)$ is shown below.

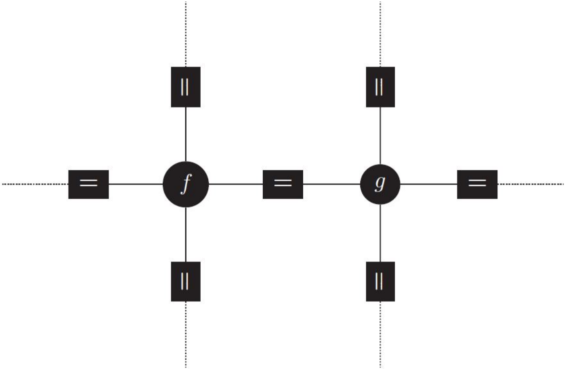


(a) $\Omega = \Omega_0$

Proof cont.

2. For each $e \in E$, we subdivide e and assign $(=_2)$ to the new vertex w. Let the resulting grid be Ω_1. Then

$$\text{Holant}(\Omega_0, \mathcal{F} \mid \mathcal{G}) = \text{Holant}(\Omega_1, \mathcal{F} \cup \mathcal{G} \cup \{=_2\})$$

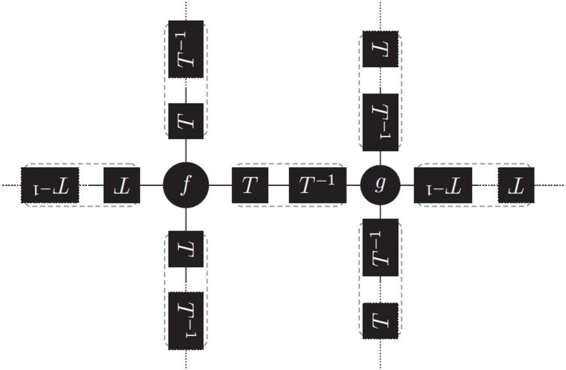


(b) Ω_1

Proof cont.

3. We subdivide w to get two vertices u', v'. Assign to u' and v' the binary signature h_u and h_v, respectively, with signature matrices T and T^{-1}, respectively. Let the resulting grid be Ω_2. Then

$$\text{Holant}(\Omega_1, \mathcal{F} \cup \mathcal{G} \cup \{=_2\}) = \text{Holant}(\Omega_2, \mathcal{F} \cup \mathcal{G} \cup \{T, T^{-1}\})$$

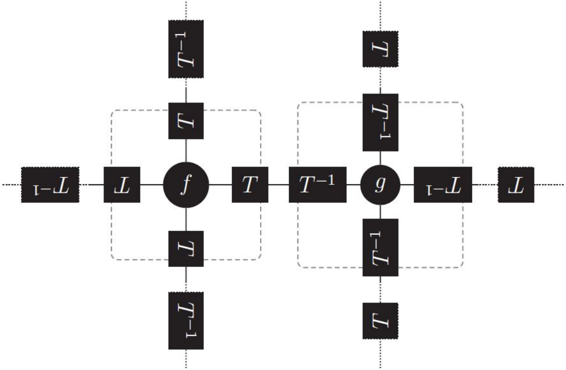


(c) Ω_2

Proof cont.

4. Ω_3 is the same as Ω_2, except we associate the binary T and T^{-1} to the original signatures f and g first. Then

$$\text{Holant}(\Omega_2, \mathcal{F} \cup \mathcal{G} \cup \{T, T^{-1}\}) = \text{Holant}(\Omega_3, \mathcal{F} \cup \mathcal{G} \cup \{T, T^{-1}\})$$



(d) Ω_3

Proof cont.

5. We contract edges (u, u') and (v', v). This defines $\Omega' = \Omega_4$.
   - The underlying graph is the initial bipartite graph G.
   - Any vertex $u \in U$ of degree $deg(u)$ assigned $f \in \mathcal{F}$ in Ω is now assigned $f \cdot T^{\otimes deg(u)}$.
   - Any vertex $v \in V$ of degree $deg(v)$ assigned $g \in \mathcal{G}$ in Ω is now assigned $(T^{-1})^{\otimes deg(v)} \cdot g$.

Then,

$$\text{Holant}(\Omega_3, \mathcal{F} \cup \mathcal{G} \cup \{T, T^{-1}\}) = \text{Holant}(\Omega_4, \mathcal{F}T \mid T^{-1}\mathcal{G}).$$

□

Overview

Overview

- Some counting problems are #P hard on general instances, but are tractable over planar structures.

- Some counting problems are #P hard on general instances, but are tractable over planar structures.
- The most representative example is counting perfect matchings.
- Counting perfect matchings in general graphs is #P-complete under Turing reductions.
- Counting perfect matchings in planar graphs can be solved by Kasteleyn's algorithm (FKT algorithm) in polynomial time.

- Some counting problems are #P hard on general instances, but are tractable over planar structures.
- The most representative example is counting perfect matchings.
- Counting perfect matchings in general graphs is #P-complete under Turing reductions.
- Counting perfect matchings in planar graphs can be solved by Kasteleyn's algorithm (FKT algorithm) in polynomial time.
- This algorithm can be extended to a *universal* strategy for a broad class of counting problems.
- These results build on the theory of matchgates and holographic algorithms.

FKT algorithm

- Kasteleyn's algorithm is a method for
  - counting perfect matchings
  - computing the weighted sum of perfect matchings

  in "Pfaffian orientable" graphs.

- Planar graphs are Pfaffian orientable.

Oddly oriented cycles

Fact

If M, M' are two perfect matchings in G, then $M \cup M'$ is a collection of single edges and even length cycles.

Oddly oriented cycles

Fact

If M, M' are two perfect matchings in G, then $M \cup M'$ is a collection of single edges and even length cycles.

An orientation $\overrightarrow{G}$ of an undirected graph G is an assignment of a direction to each of its edges.

Oddly oriented cycles

Fact

If M, M' are two perfect matchings in G, then $M \cup M'$ is a collection of single edges and even length cycles.

An orientation $\overrightarrow{G}$ of an undirected graph G is an assignment of a direction to each of its edges.

Definition 1

Let $G = (V, E)$ be an undirected graph, and $\overrightarrow{G}$ an orientation of G.
We say that an even cycle C of G is oddly oriented by $\overrightarrow{G}$, if when traversing C, in either direction, the number of co-oriented edges is odd.

Pfaffian orientation

Definition 2

An orientation $\overrightarrow{G}$ of G is Pfaffian if the following condition holds:
for any two perfect matchings M, M' in G, every cycle in $M \cup M'$ is oddly oriented by $\overrightarrow{G}$.