CS 640 Course Presentation

Undirected Connectivity in Log Space - O. Reingold

Keerti Choudhary

Contents

Background

Background

- SL is class of problems solvable by symmetric non-deterministic TM in log space.

Background

- SL is class of problems solvable by symmetric non-deterministic TM in log space.

- $L \subseteq SL \subseteq NL$.

Background

- SL is class of problems solvable by symmetric non-deterministic TM in log space.
- $L \subseteq SL \subseteq NL$.
- USTCON (Undirected s-t connectivity) was proved to be complete for class SL.

Background

- SL is class of problems solvable by symmetric non-deterministic TM in log space.
- L $\subseteq$ SL $\subseteq$ NL.
- USTCON (Undirected s-t connectivity) was proved to be complete for class SL.
- This paper proves that USTCON can be solved in log space, thus shows L=SL.

Main Idea

Main Idea

- Suppose a undirected graph G has max degree bounded by ‘D’ and diameter bounded by ‘ρ’.

Main Idea

- Suppose a undirected graph G has max degree bounded by 'D' and diameter bounded by 'ρ'.
- Then s-t connectivity can be solved in $O(\rho \log D)$ space.

Main Idea

- Suppose a undirected graph G has max degree bounded by 'D' and diameter bounded by 'ρ'.
- Then s-t connectivity can be solved in $O(\rho \log D)$ space.
- So, we will convert G to a new graph satisfying
  - degree bounded by some constant c
  - diameter is $O(\log n)$.

Expander Graphs

Expander Graphs

- A D-regular graph is ε-expander if any set S of size less than n/2 has closed neighbourhood of size at least $(1+\varepsilon)|S|$.

Expander Graphs

- A D-regular graph is ε-expander if any set S of size less than n/2 has closed neighbourhood of size at least $(1+\varepsilon)|S|$.

- Diameter of such graphs is bounded by $2\log_{1+\varepsilon} n$.

Expander Graphs

- A D-regular graph is ε-expander if any set S of size less than n/2 has closed neighbourhood of size at least $(1+\varepsilon)|S|$.
- Diameter of such graphs is bounded by $2\log_{1+\varepsilon} n$.
- Idea is to convert G to an expander graph whose degree is also bounded.

Expander Graphs

Expander Graphs

- Regular graphs with small second largest eigen value (λ) of normalised adjacency matrix.

Expander Graphs

- Regular graphs with small second largest eigen value (λ) of normalised adjacency matrix.

- For every D, λ (λ<1), there exist ε>0 such that all D-regular graphs with second largest eigen value bounded by λ are ε-expander.

Expander Graphs

- Regular graphs with small second largest eigen value (λ) of normalised adjacency matrix.

- For every D, λ (λ<1), there exist ε>0 such that all D-regular graphs with second largest eigen value bounded by λ are ε-expander.

- To convert G to an expander graph we need to bound its λ .

Powering

Powering

- Squaring of the graph reduces the second largest eigen value to λ^2 .

Powering

- Squaring of the graph reduces the second largest eigen value to λ^2 .
- As $\lambda(G) < 1 - 1/DN^2$, so squaring the graph $O(\log N)$ times would bound λ .

Powering

- Squaring of the graph reduces the second largest eigen value to λ^2 .
- As $\lambda(G) < 1 - 1/DN^2$, so squaring the graph $O(\log N)$ times would bound λ .
- But the cost is that degree will also be squared.

Bounding the degree

Bounding the degree

Two methods -

Bounding the degree

Two methods -

1. Replacement Product

Bounding the degree

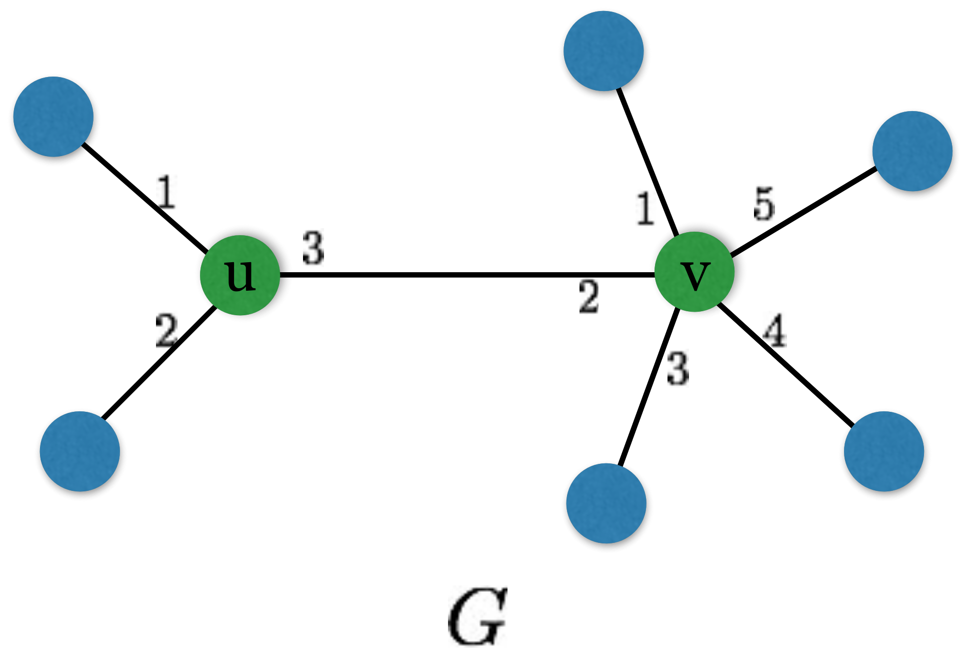
Two methods -

1. Replacement Product
2. Zig-zag Product

Rotation Map

In graph G, $Rot_G(u, i) = (v, j)$ iff,

- i^{th} vertex in adjacency list of u is v, and
- j^{th} vertex in adjacency list of v is u.



$$Rot_G(u, 3) = (v, 2)$$

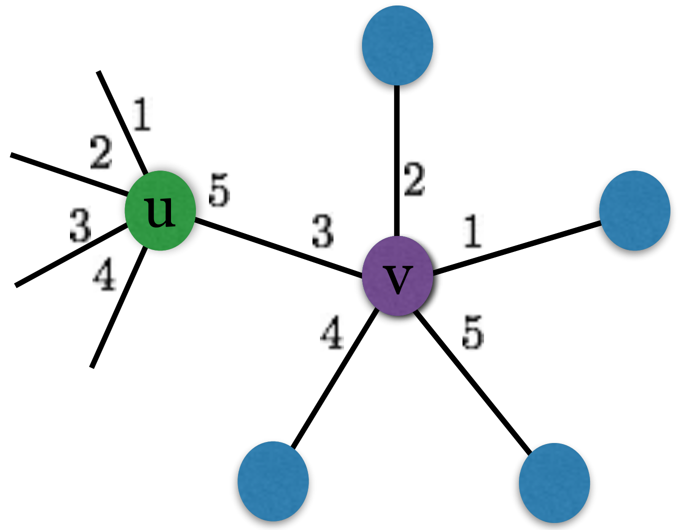
$$Rot_G(v, 2) = (u, 3)$$

Replacement Product

Replacement Product

- Given a D-regular graph G, and a d-regular (d<<D) connected graph H on D vertices. Replace each vertex of G by H.

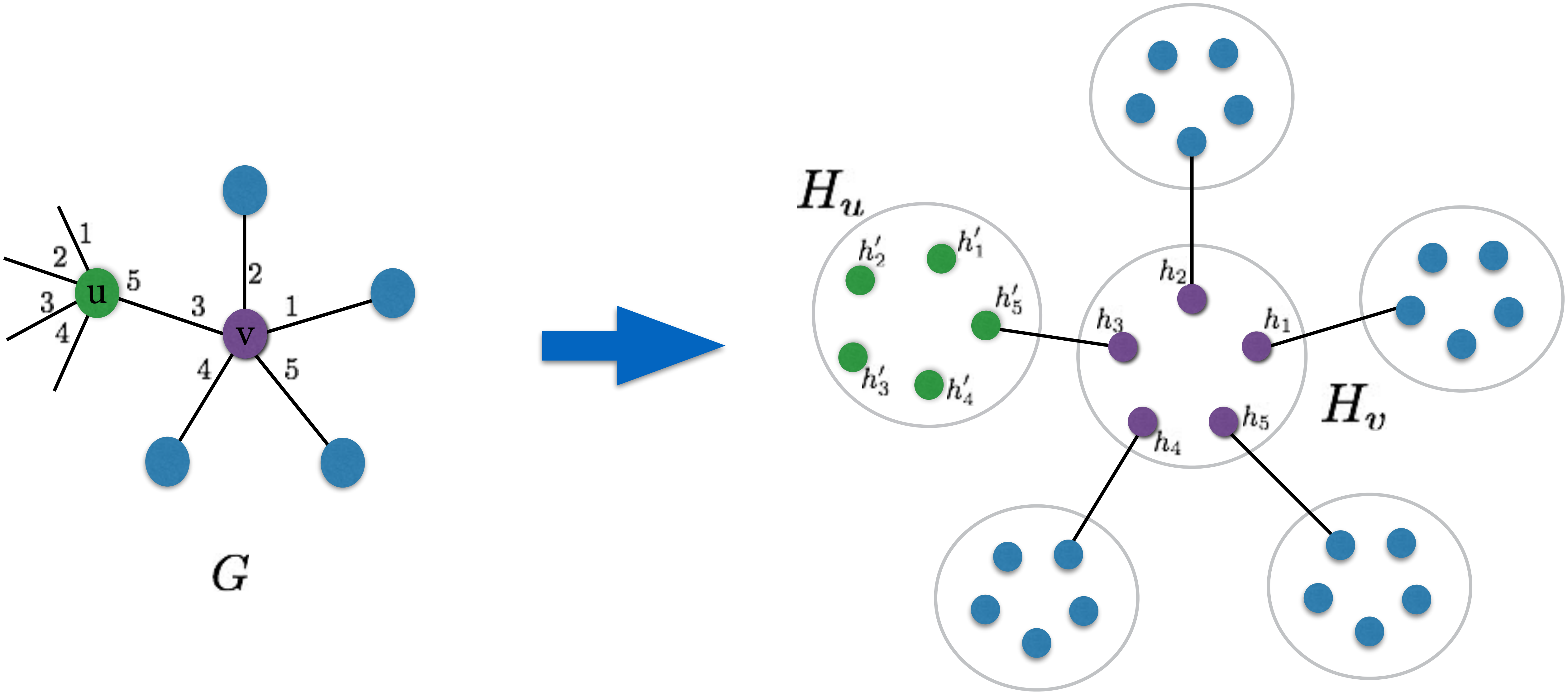
Replacement Product



G

$$Rot_G(u, 5) = (v, 3)$$

Replacement Product



$$Rot_G(u, 5) = (v, 3)$$

Replacement Product

- Given a D-regular graph G, and a d-regular (d<<D) connected graph H on D vertices. Replace each vertex of G by H.

- New graph is a d+1 regular.

Zig-zag Product

Zig-zag Product

(a,d) is an edge in $G \circledz H$ iff a—b—c—d is a path in replacement product satisfying -

Zig-zag Product

(a,d) is an edge in $G \textcircled{z} H$ iff a—b—c—d is a path in replacement product satisfying -

- (a,b) and (c,d) are edges in different instances of H

Zig-zag Product

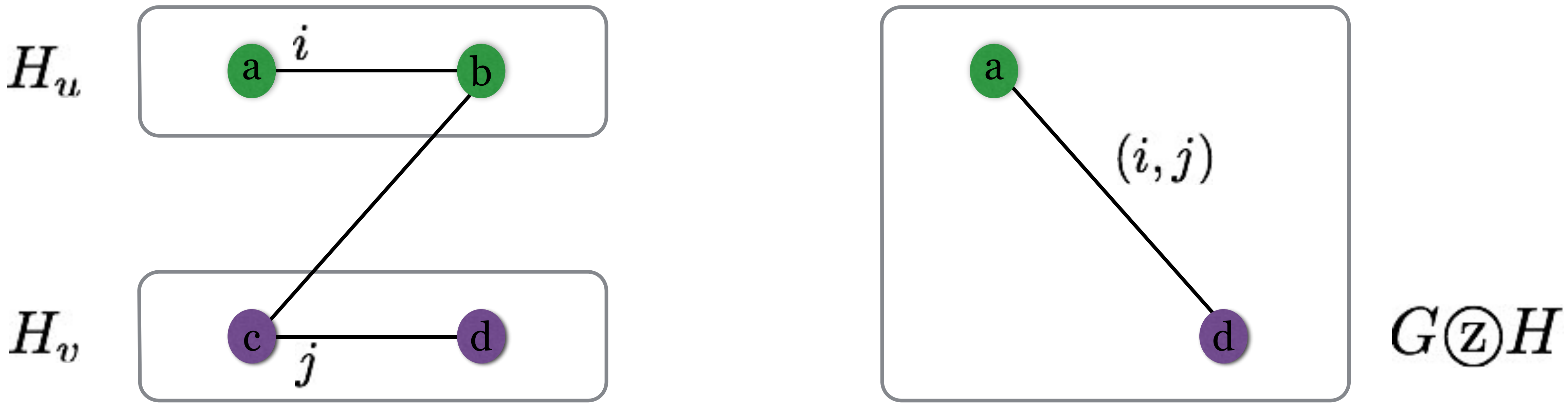
(a,d) is an edge in $G \circledz H$ iff a—b—c—d is a path in replacement product satisfying -

- (a,b) and (c,d) are edges in different instances of H
- (b,c) corresponds to edge of G.

Zig-zag Product

(a,d) is an edge in $G \textcircled{z} H$ iff a—b—c—d is a path in replacement product satisfying -

- (a,b) and (c,d) are edges in different instances of H
- (b,c) corresponds to edge of G.



Zig-zag Product

Zig-zag Product

- So, if G is D-regular and H is d-regular than, $G \textcircled{z} H$ is d^2-regular.

Zig-zag Product

- So, if G is D-regular and H is d-regular than, $G \textcircled{z} H$ is d^2-regular.

- THEOREM - If G is an (N, D, λ)-graph and H is a (D, d, α)-graph, then

$$\lambda(G \textcircled{z} H) \leq 1 - \frac{1}{2}(1 - \alpha^2)(1 - \lambda)$$

Zig-zag Product

- So, if G is D-regular and H is d-regular than, $G ⓩ H$ is d^2-regular.
- THEOREM - If G is an (N, D, λ)-graph and H is a (D, d, α)-graph, then

$$\lambda(G ⓩ H) \leq 1 - \frac{1}{2}(1 - \alpha^2)(1 - \lambda)$$

- Here λ of the new graph is bounded above.

Main Transformation

Main Transformation

- G_0 is non-bipartite D^{16}-regular graph on N vertices, H is D-regular graph on D^{16} vertices. Also, $\lambda(H) = \frac{1}{2}$.

Main Transformation

- G_0 is non-bipartite D^{16}-regular graph on N vertices, H is D-regular graph on D^{16} vertices. Also, $\lambda(H) = \frac{1}{2}$.
- Let $l = 2\lceil log(DN^2)\rceil$. For i=1 to l, define

$$G_i = (G_{i-1} ⓩ H)^8$$

Main Transformation

- G_0 is non-bipartite D^{16}-regular graph on N vertices, H is D-regular graph on D^{16} vertices. Also, $\lambda(H) = \frac{1}{2}$.
- Let $l = 2\lceil log(DN^2) \rceil$. For i=1 to l, define

$$G_i = (G_{i-1} \text{ⓩ} H)^8$$

- Then, $\lambda(G_l) < \frac{1}{2}$.

Main Transformation

- G_0 is non-bipartite D^{16}-regular graph on N vertices, H is D-regular graph on D^{16} vertices. Also, $\lambda(H) = \frac{1}{2}$.

- Let $l = 2\lceil log(DN^2)\rceil$. For i=1 to l, define

$$G_i = (G_{i-1} \circledz H)^8$$

- Then, $\lambda(G_l) < \frac{1}{2}$.

- Proof Idea :

Main Transformation

- G_0 is non-bipartite D^{16}-regular graph on N vertices, H is D-regular graph on D^{16} vertices. Also, $\lambda(H) = \frac{1}{2}$.
- Let $l = 2\lceil log(DN^2)\rceil$. For i=1 to l, define

$$G_i = (G_{i-1} \textcircled{z} H)^8$$

- Then, $\lambda(G_l) < \frac{1}{2}$.
- Proof Idea : - $\lambda(G_0) < 1 - 1/D^{16}N^2$

Main Transformation

- G_0 is non-bipartite D^{16}-regular graph on N vertices, H is D-regular graph on D^{16} vertices. Also, $\lambda(H) = \frac{1}{2}$.
- Let $l = 2\lceil log(DN^2)\rceil$. For i=1 to l, define

$$G_i = (G_{i-1} \text{ⓩ} H)^8$$

- Then, $\lambda(G_l) < \frac{1}{2}$.
- Proof Idea :
  - $\lambda(G_0) < 1 - 1/D^{16}N^2$
  - $\lambda(G_i) < max\{\lambda(G_{i-1})^2, 1/2\}$

Main Transformation

Converting G to regular graph G_0 of size N= n^2 :

Main Transformation

Converting G to regular graph G_0 of size N= n^2 :

- Each vertex (u) is replaced by a n-cycle ($u_1, u_2, \ldots, u_n$).

Main Transformation

Converting G to regular graph G_0 of size N= n^2 :

- Each vertex (u) is replaced by a n-cycle ($u_1, u_2, ..., u_n$).
- If $Rot_G(u, i) = (v, j)$, then (u_i, v_j) is an edge in G_0.

Main Transformation

Converting G to regular graph G_0 of size N= n^2 :

- Each vertex (u) is replaced by a n-cycle ($u_1, u_2, ..., u_n$).
- If $Rot_G(u, i) = (v, j)$, then (u_i, v_j) is an edge in G_0.
- Multiple loops are added to each vertex in G_0 until its degree reaches D^{16}.

Main Transformation

Converting G to regular graph G_0 of size N= n^2 :

- Each vertex (u) is replaced by a n-cycle ($u_1, u_2, ..., u_n$).
- If $Rot_G(u,i) = (v,j)$, then (u_i, v_j) is an edge in G_0.
- Multiple loops are added to each vertex in G_0 until its degree reaches D^{16}.

H is of constant size and can be encoded in TM.

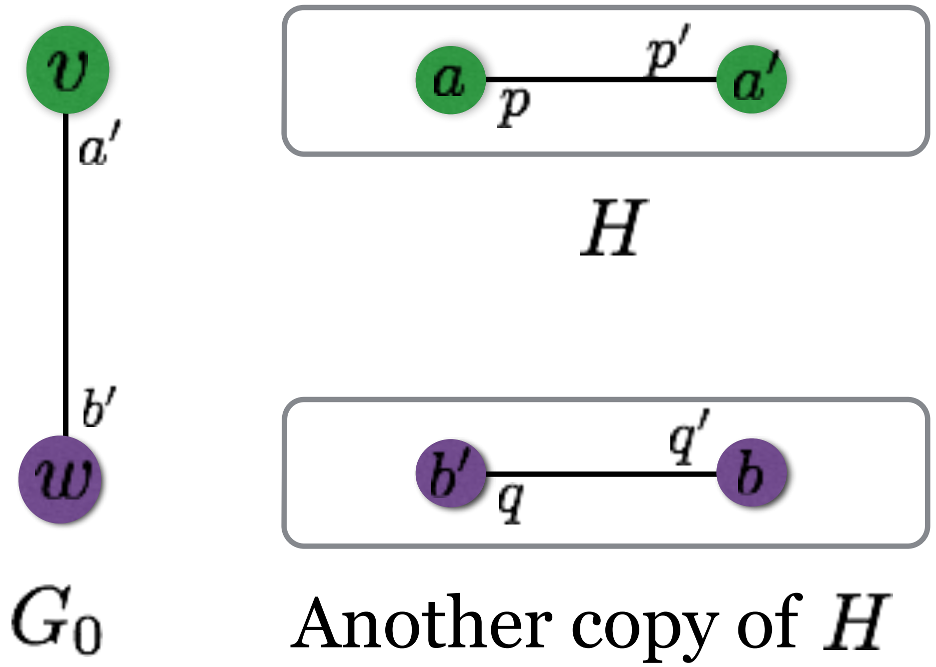
Implementation

Implementation

Simpler Case : $G_0 \textcircled{z} H$

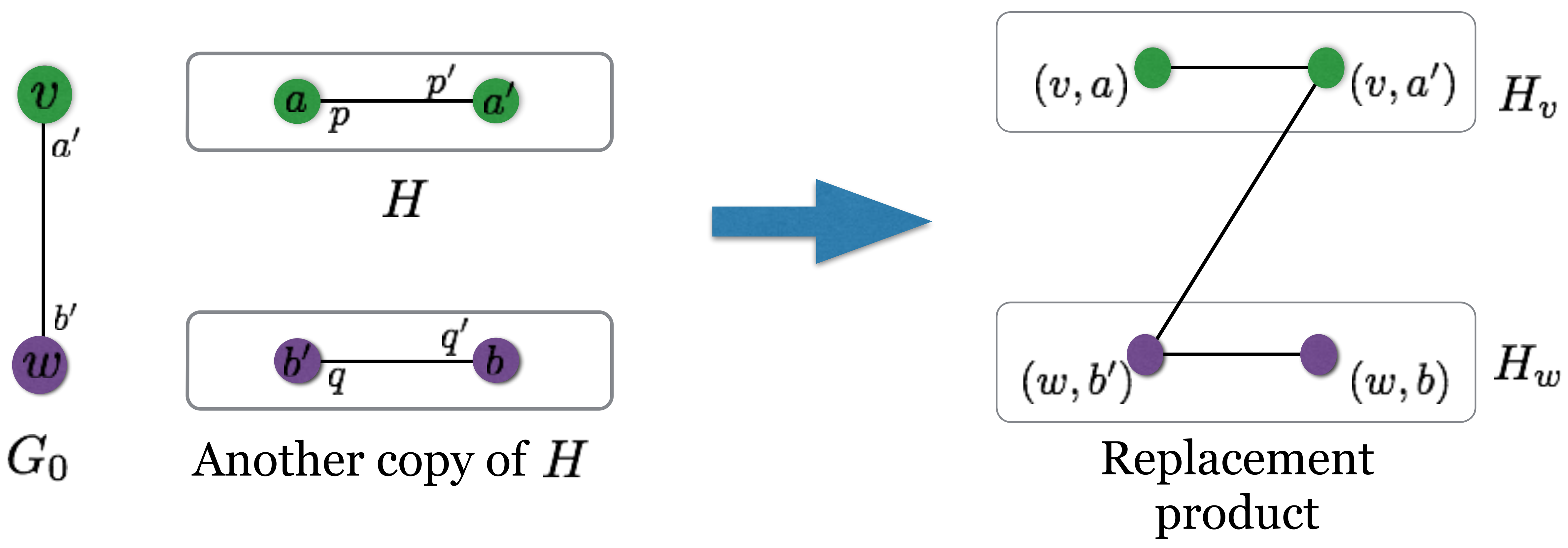
Implementation

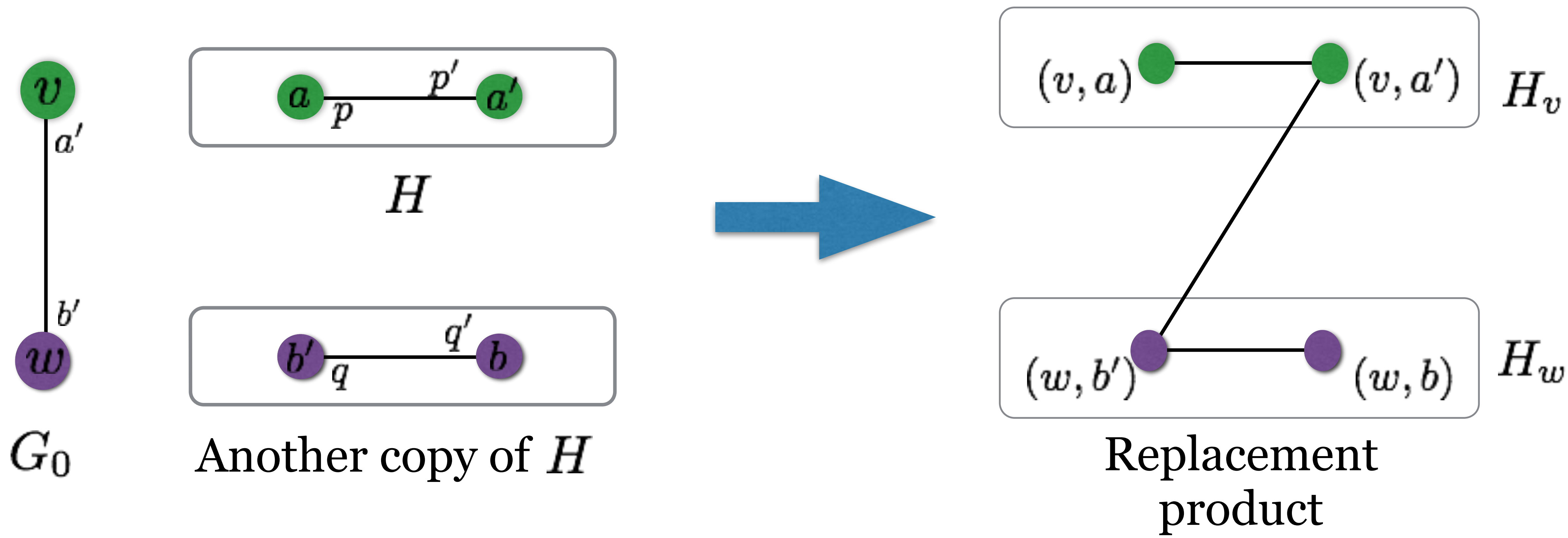
Simpler Case : $G_0 \textcircled{z} H$



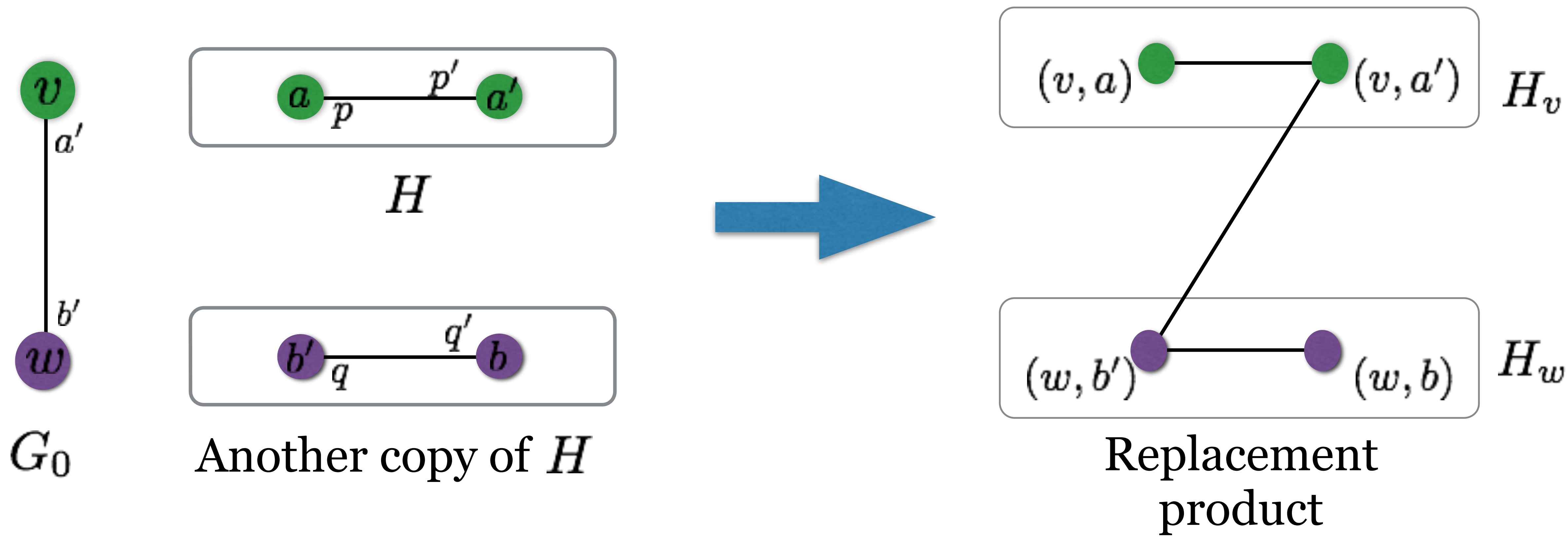
Implementation

Simpler Case : $G_0 ⓩ H$



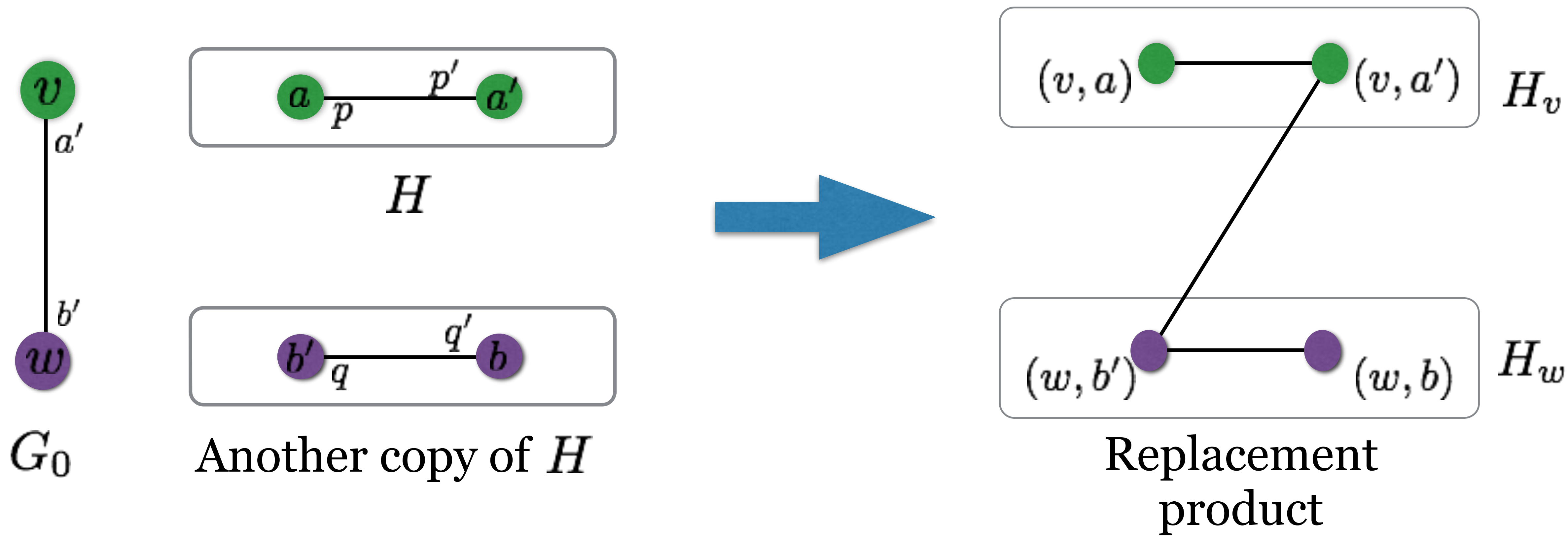


Now, $(v, a) \in [N] \times D^{16}$ and, we want to compute its $(p, q)^{th}$ neighbour



Now, $(v,a) \in [N] \times D^{16}$ and,
we want to compute its $(p,q)^{th}$
neighbour

$$((v,a),(p,q))$$



Now, $(v, a) \in [N] \times D^{16}$ and, we want to compute its $(p, q)^{th}$ neighbour

$$((v, a), (p, q))$$

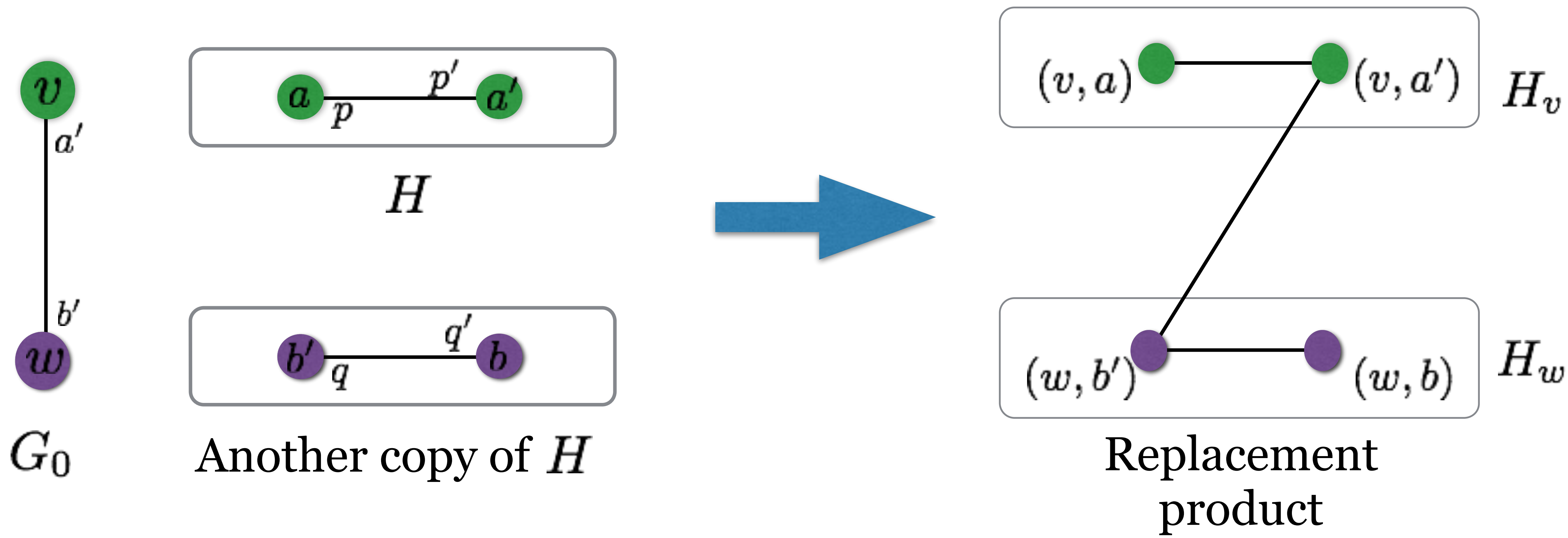
$Rot_{G_0 \textcircled{z} H}((v, a), (p, q))$:

$(a, p) \leftarrow Rot_H(a, p)$

$(v, a) \leftarrow Rot_{G_0}(v, a)$

$(a, q) \leftarrow Rot_H(a, q)$

Swap (p, q)



Now, $(v, a) \in [N] \times D^{16}$ and, we want to compute its $(p, q)^{th}$ neighbour

$$((v, a'), (p', q))$$

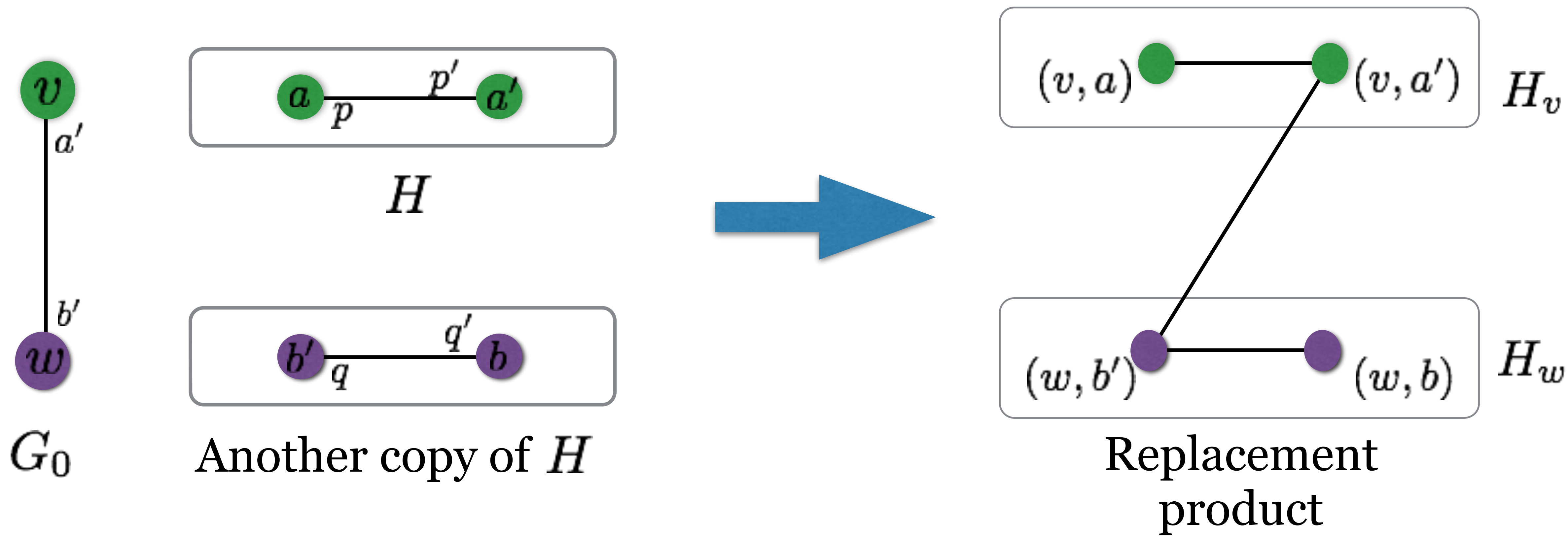
$Rot_{G_0 \textcircled{z} H}((v, a), (p, q))$:

$(a, p) \leftarrow Rot_H(a, p)$

$(v, a) \leftarrow Rot_{G_0}(v, a)$

$(a, q) \leftarrow Rot_H(a, q)$

Swap (p, q)



Now, $(v, a) \in [N] \times D^{16}$ and, we want to compute its $(p, q)^{th}$ neighbour

$$((w, b'), (p', q))$$

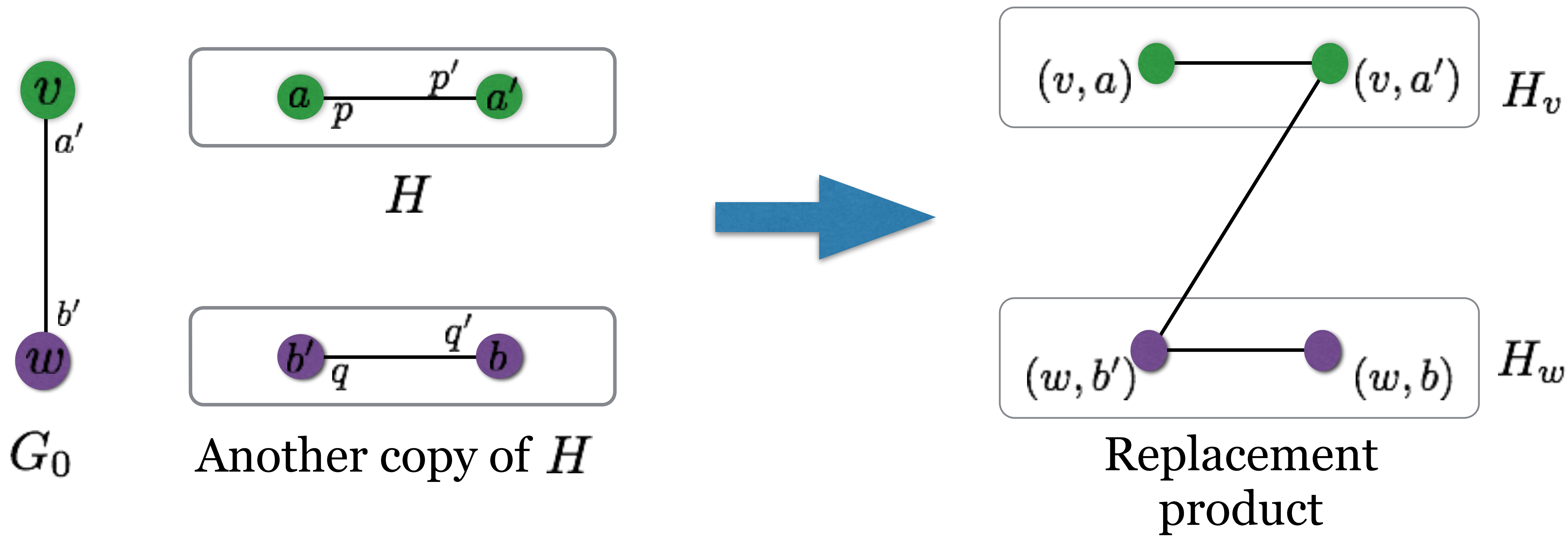
$Rot_{G_0 \textcircled{z} H}((v, a), (p, q))$:

$(a, p) \leftarrow Rot_H(a, p)$

$(v, a) \leftarrow Rot_{G_0}(v, a)$

$(a, q) \leftarrow Rot_H(a, q)$

Swap (p, q)



Now, $(v, a) \in [N] \times D^{16}$ and, we want to compute its $(p, q)^{th}$ neighbour

$$((w, b), (p', q'))$$

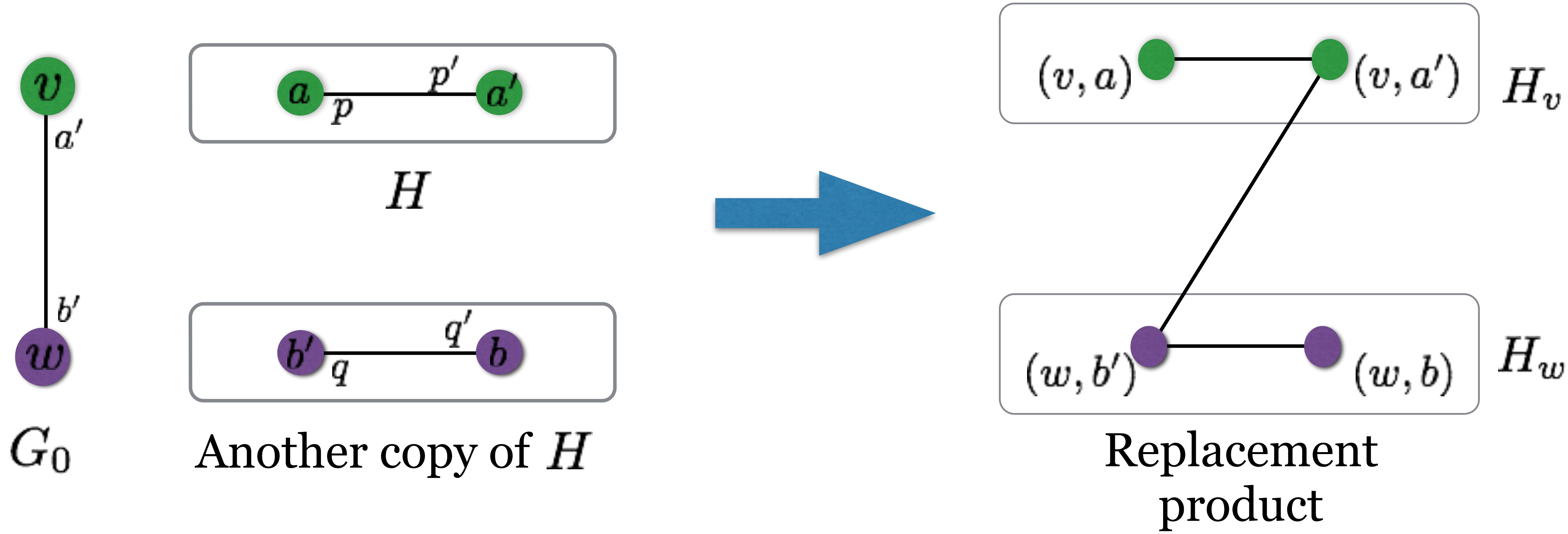
$Rot_{G_0 \textcircled{z} H}((v, a), (p, q))$:

$(a, p) \leftarrow Rot_H(a, p)$

$(v, a) \leftarrow Rot_{G_0}(v, a)$

$(a, q) \leftarrow Rot_H(a, q)$

Swap (p, q)



Now, $(v,a) \in [N] \times D^{16}$ and, we want to compute its $(p,q)^{th}$ neighbour

$$((w,b),(q',p'))$$

$Rot_{G_0 ⓩ H}((v,a),(p,q))$:

$(a,p) \leftarrow Rot_H(a,p)$

$(v,a) \leftarrow Rot_{G_0}(v,a)$

$(a,q) \leftarrow Rot_H(a,q)$

Swap (p,q)

Implementation

Implementation

- G_l will be of size $N \times (D^{16})^l$, and any vertex

$$\overline{v} \in [N] \times ([D^{16}])^l$$

Implementation

- G_l will be of size $N \times (D^{16})^l$, and any vertex

$$\overline{v} \in [N] \times ([D^{16}])^l$$

- We need to compute the rotation map in log space.

Implementation

Main Algorithm

$Rot_{G_i}(v, a_0, ..., a_i)$:

1. $(p_1, q_1, p_2, ..., p_8, q_8) \leftarrow a_i$
2. For j=1 to 8
   - $a_{i-1}, p_j \leftarrow Rot_H(a_{i-1}, p_j)$
   - $v, a_0, ..., a_{i-1} \leftarrow Rot_{G_{i-1}}((v, a_0, ..., a_{i-2}), a_{i-1})$
   - $a_{i-1}, q_j \leftarrow Rot_H(a_{i-1}, q_j)$
3. $a_i \leftarrow (q_8, p_8, ..., p_2, q_1, p_1)$

THANK YOU