On Token Sliding (Reconfiguration) Graphs of Independent Sets

Duc A. Hoang

VNU University of Science, Hanoi, Vietnam
hoanganhduc@hus.edu.vn

Based on joint works with David Avis (Kyoto University, Japan)

The Korea-Taiwan-Vietnam Joint Meeting on Discrete Geometry and Geometric Measure Theory
(VIASM, Hanoi, Vietnam)

July 17–19, 2023

Outline

A Geometric Example

A Geometric Example

- *Flip graphs* are very important in computational geometry and have many applications in various settings, including
  - measuring the "similarity" (or "distance") between two geometric objects by constraining the family of "allowable flips";
  - enumerating geometric objects;
  - computing geometric objects with respect to some optimization criteria;
  - and so on

A Geometric Example

- *Flip graphs* are very important in computational geometry and have many applications in various settings, including
  - measuring the "similarity" (or "distance") between two geometric objects by constraining the family of "allowable flips";
  - enumerating geometric objects;
  - computing geometric objects with respect to some optimization criteria;
  - and so on
- For a given graph G and a fixed integer $k \geq 1$, the TS_k*-(reconfiguration) graph* (or TS_k*-graph*) of G, denoted by $\mathsf{TS}_k(G)$, is the graph whose
  - *vertices* are *independent sets* (or *stable sets*, which are vertex subsets of pairwise non-adjacent vertices) of G
    - Each stable set can be seen as a *set of tokens placed on vertices of G*
  - *edges* are defined by *Token Sliding* (TS)

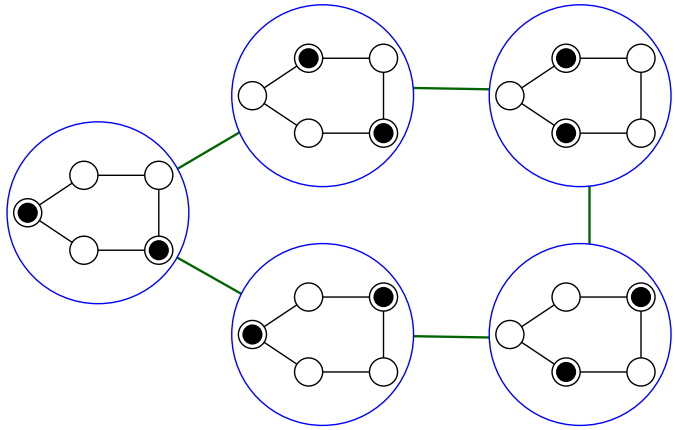


Figure: $\mathsf{TS}_2(C_5)$

A Geometric Example

- *Flip graphs* are very important in computational geometry and have many applications in various settings, including
  - measuring the "similarity" (or "distance") between two geometric objects by constraining the family of "allowable flips";
  - enumerating geometric objects;
  - computing geometric objects with respect to some optimization criteria;
  - and so on
- For a given graph G and a fixed integer $k \geq 1$, the TS_k*-(reconfiguration) graph* (or TS_k*-graph*) of G, denoted by $\mathsf{TS}_k(G)$, is the graph whose
  - *vertices* are *independent sets* (or *stable sets*, which are vertex subsets of pairwise non-adjacent vertices) of G
    - Each stable set can be seen as a *set of tokens placed on vertices of G*
  - *edges* are defined by *Token Sliding* (TS)

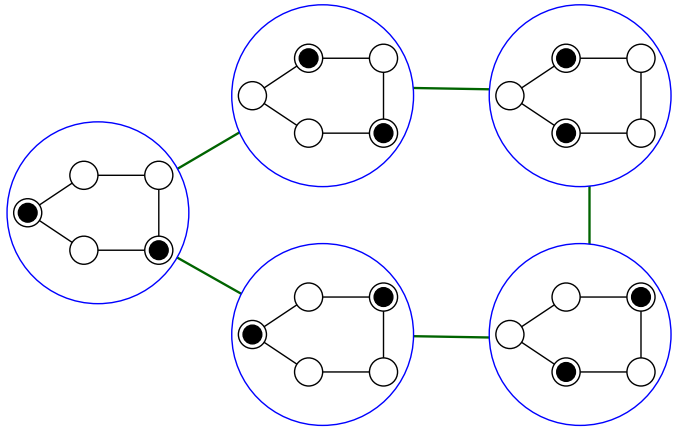


Figure: $\mathsf{TS}_2(C_5)$

- We present an example showing *a relationship between some flip graphs of triangulations and* TS_k*-graphs (of some graphs)*

A Geometric Example

- Given a set P of n points in the plane, no three collinear and no four co-circular

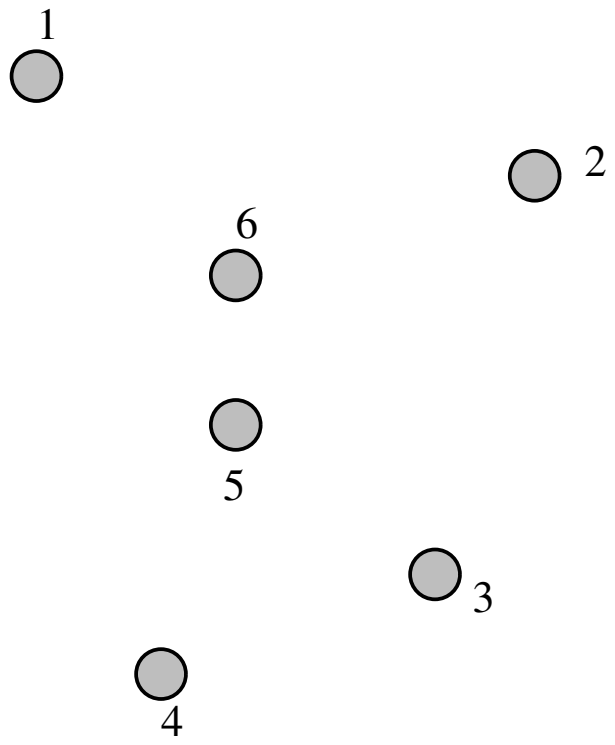


Figure: A set P of n points in the plane.

A Geometric Example

- Given a set P of n points in the plane, no three collinear and no four co-circular
- Two line segments are *intersecting* if they cross each other at an interior point of each segment, and *non-intersecting* otherwise

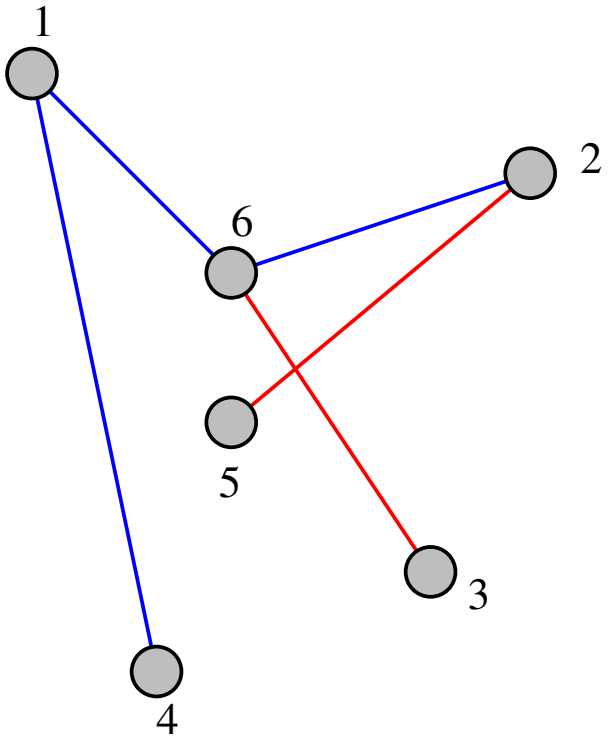


Figure: A set P of n points in the plane. Red (Blue) segments are pairwise *intersecting* (*non-intersecting*).

A Geometric Example

- Given a set P of n points in the plane, no three collinear and no four co-circular
- Two line segments are *intersecting* if they cross each other at an interior point of each segment, and *non-intersecting* otherwise
- A *triangulation* of P is any maximal set of pairwise non-intersecting segments. It is well-known that *all triangulations have the same number of edges (segments)*

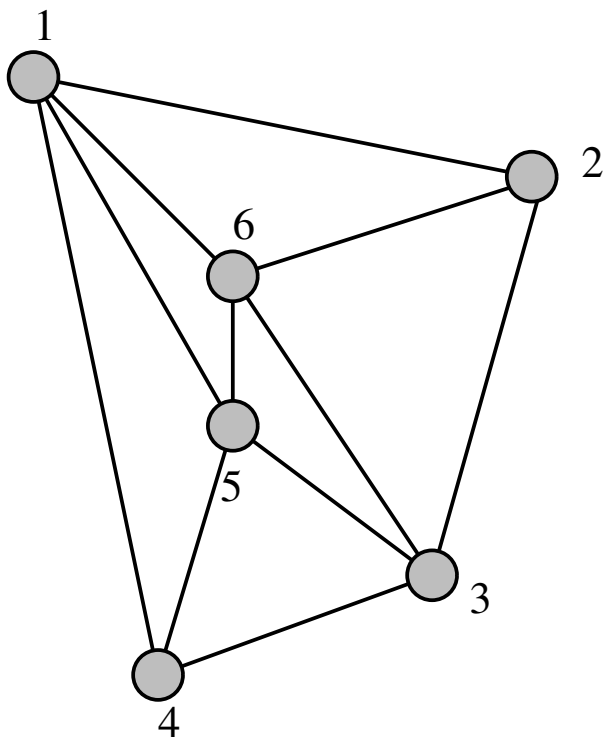


Figure: A set P of n points in the plane. A triangulation of P where each segment is colored in black

A Geometric Example

- The *edge-intersection graph G of P* is the graph whose
  - *vertices* are *the line segments with endpoints in P that intersect at least one other segment*

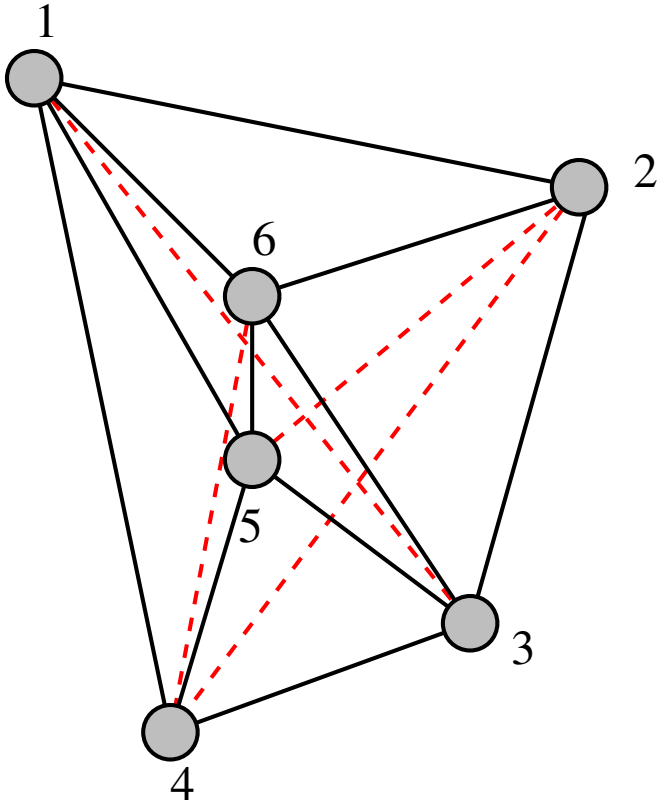


Figure: A triangulation of P is marked by black edges. The remaining line segments are marked by dashed red edges

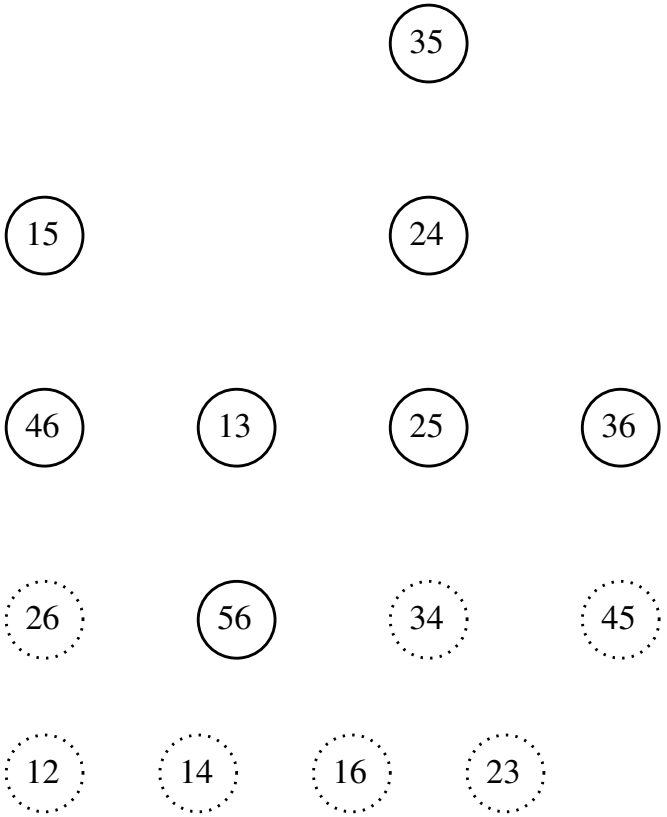


Figure: The edge-intersection graph G of P. Vertices (segments) not in G are depicted by dotted circles.

A Geometric Example

- The *edge-intersection graph G of P* is the graph whose
  - *vertices* are *the line segments with endpoints in P that intersect at least one other segment*
  - *edges* are defined between two vertices whose *corresponding line segments are intersecting*

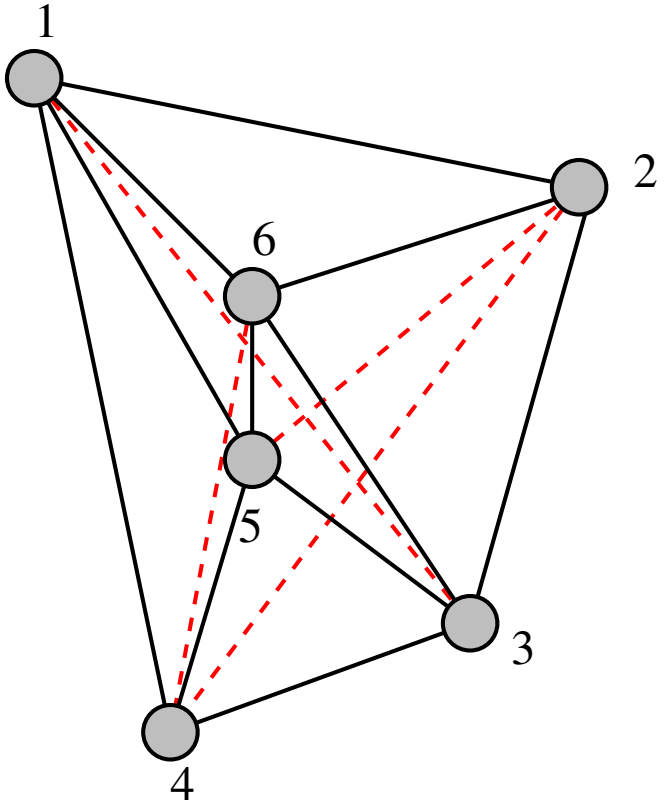


Figure: A triangulation of P is marked by black edges. The remaining line segments are marked by dashed red edges

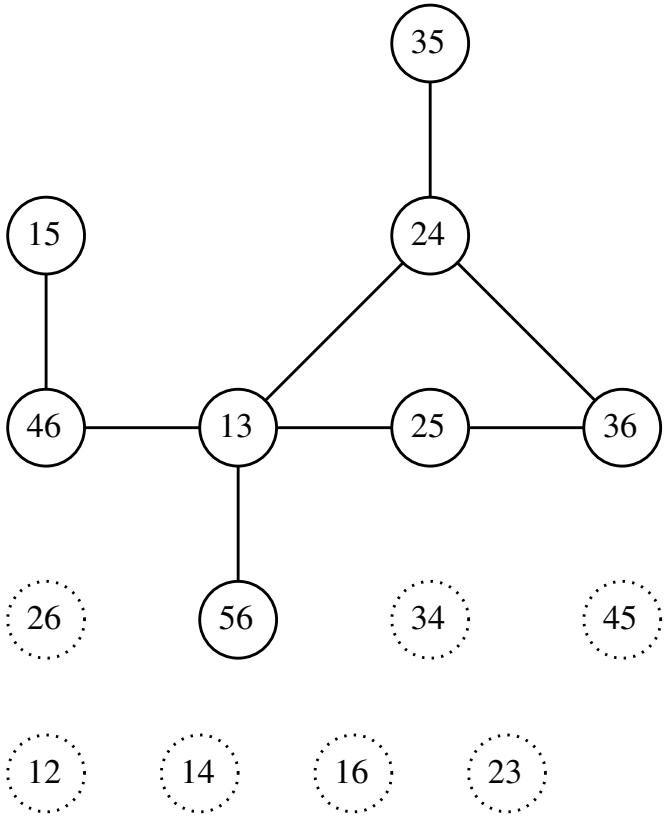


Figure: The edge-intersection graph G of P. Vertices (segments) not in G are depicted by dotted circles.

A Geometric Example

- The *flip graph of triangulations of P* is the graph whose
  - *vertices* are *triangulations of P*
  - *edges* are defined via the *flip operation*: two triangulations T_1, T_2 of P are adjacent if one can be obtained from the other by *flipping* the diagonal of a convex quadrilateral

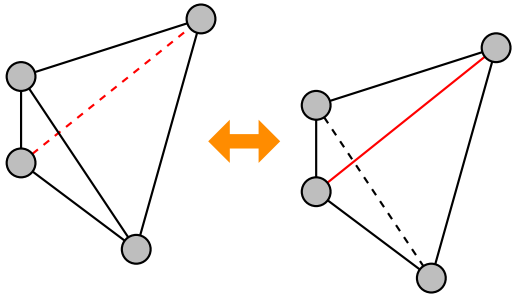


Figure: The flip operation

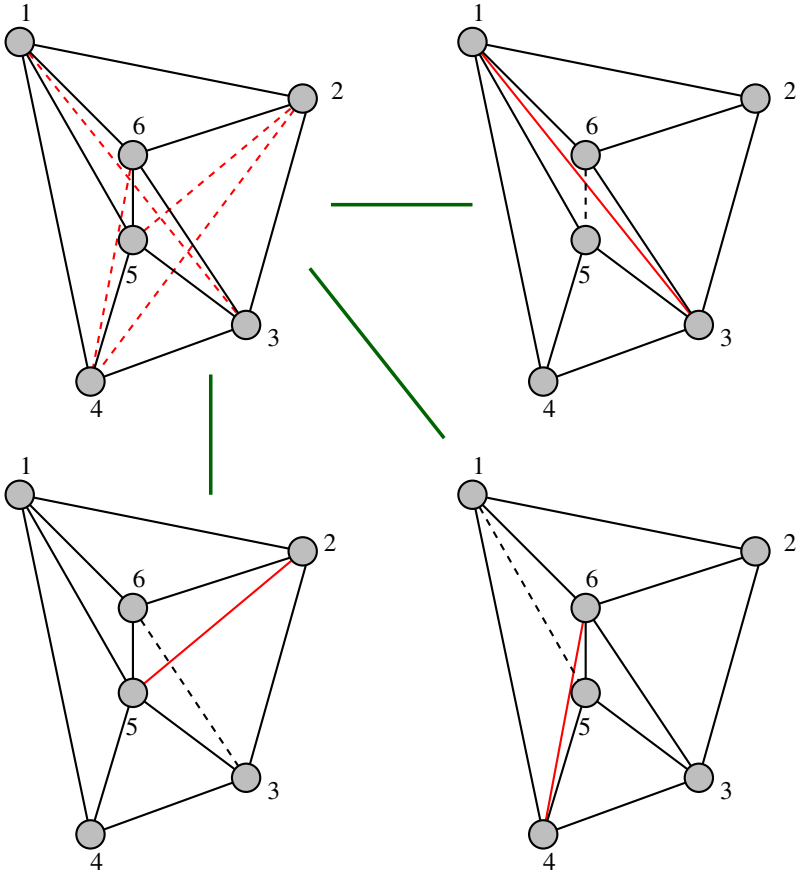
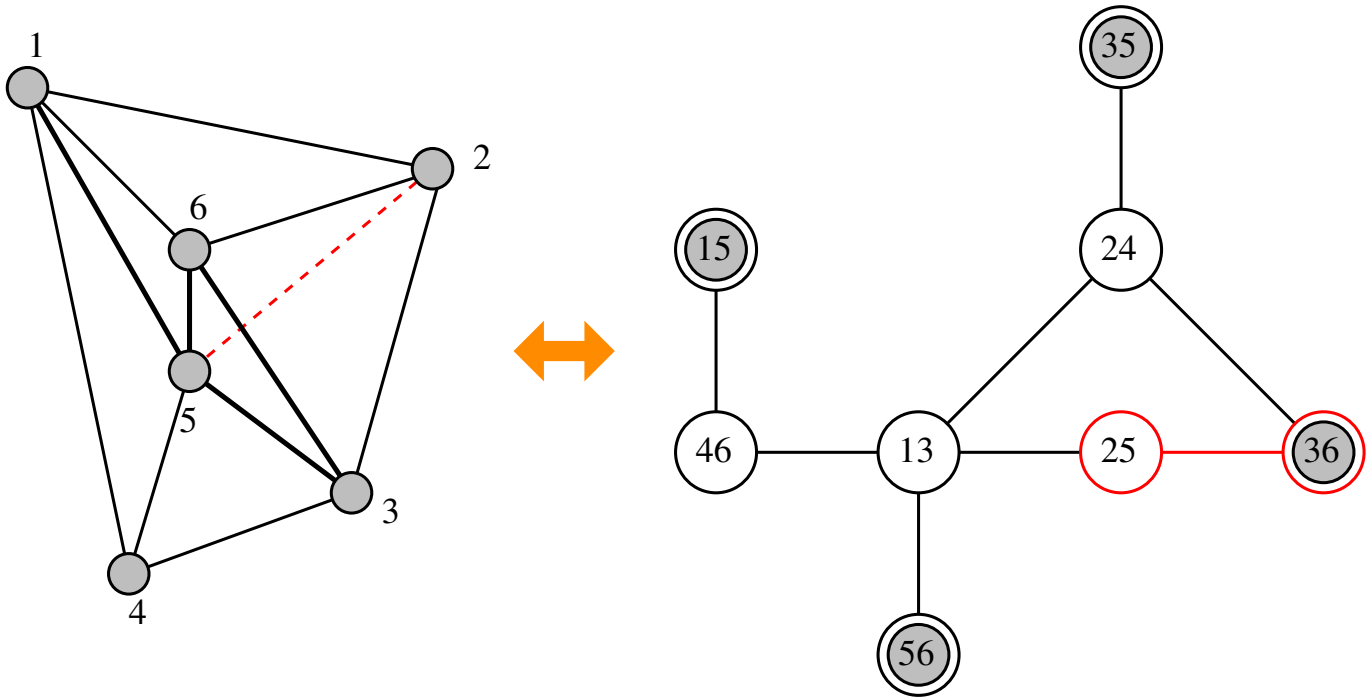


Figure: A vertex (triangulation) in the flip graph of triangulations of P (top left) and its adjacent ones

A Geometric Example

- A set of k pairwise non-intersecting segments where each member intersects at least one other segment $\Leftrightarrow$ A vertex of $\mathsf{TS}_k(G)$ (e.g., size-k stable set of G)
- An edge of the flip graph of triangulations of P $\Leftrightarrow$ An edge of $\mathsf{TS}_k(G)$

Flip graph of triangulations of P $\Leftrightarrow$ $\mathsf{TS}_{\alpha(G)}$-graph of the edge-intersection graph G of P



Note: $\alpha(G)$ denotes the size of a maximum stable set of G

TS_k-Graphs: Background and Motivation

TS_k-Graphs: Background and Motivation

- Related works involve *Token Graphs*
  - A *k-token graph* $F_k(G)$ of a graph G is the graph whose
    - *vertices* are size-k vertex subsets of G
    - *edges* are defined under Token Sliding

TS_k-Graphs: Background and Motivation

- Related works involve *Token Graphs*
  - A *k-token graph* $F_k(G)$ of a graph G is the graph whose
    - *vertices* are size-k vertex subsets of G
    - *edges* are defined under Token Sliding
  - One of the earliest results is [Alavi, Behzad, Erdős, and Lick 1991]
    - They studied the regularity, bipartitedness, Eulerianity, etc. of $F_2(G)$—which they called the *double vertex graphs*

TS_k-Graphs: Background and Motivation

- Related works involve *Token Graphs*
  - A *k-token graph* $F_k(G)$ of a graph G is the graph whose
    - *vertices* are size-k vertex subsets of G
    - *edges* are defined under Token Sliding
  - One of the earliest results is [Alavi, Behzad, Erdős, and Lick 1991]
    - They studied the regularity, bipartitedness, Eulerianity, etc. of $F_2(G)$—which they called the *double vertex graphs*
  - *Combinatorial and algebraic properties* of $F_k(G)$ have been *extensively studied*, especially since it was *reintroduced in [Fabila-Monroy, Flores-Peñaloza, et al. 2012]*, e.g., see [Lew 2023]; [Fabila-Monroy and Trujillo-Negrete 2023]

TS_k-Graphs: Background and Motivation

- Related works involve *Token Graphs*
  - A *k-token graph* $F_k(G)$ of a graph G is the graph whose
    - *vertices* are size-k vertex subsets of G
    - *edges* are defined under Token Sliding
  - One of the earliest results is [Alavi, Behzad, Erdős, and Lick 1991]
    - They studied the regularity, bipartitedness, Eulerianity, etc. of $F_2(G)$—which they called the *double vertex graphs*
  - *Combinatorial and algebraic properties* of $F_k(G)$ have been *extensively studied*, especially since it was *reintroduced in [Fabila-Monroy, Flores-Peñaloza, et al. 2012]*, e.g., see [Lew 2023]; [Fabila-Monroy and Trujillo-Negrete 2023]
- General Framework: *Reconfiguration Problems*
  - A *growing research area* in computer science which primarily aims to *understand the solution space of a computational problem*

TS_k-Graphs: Background and Motivation

- Related works involve *Token Graphs*
  - A *k-token graph* $F_k(G)$ of a graph G is the graph whose
    - *vertices* are size-k vertex subsets of G
    - *edges* are defined under Token Sliding
  - One of the earliest results is [Alavi, Behzad, Erdős, and Lick 1991]
    - They studied the regularity, bipartitedness, Eulerianity, etc. of $F_2(G)$—which they called the *double vertex graphs*
  - *Combinatorial and algebraic properties* of $F_k(G)$ have been *extensively studied*, especially since it was *reintroduced in [Fabila-Monroy, Flores-Peñaloza, et al. 2012]*, e.g., see [Lew 2023]; [Fabila-Monroy and Trujillo-Negrete 2023]
- General Framework: *Reconfiguration Problems*
  - A *growing research area* in computer science which primarily aims to *understand the solution space of a computational problem*
  - Under the "reconfiguration setting", $\mathsf{TS}_k(G)$ has been *extremely well-studied* from the *algorithmic perspective* [van den Heuvel 2013]; [Nishimura 2018]; [Bousquet, Mouawad, Nishimura, and Siebertz 2022]

TS_k-Graphs: Background and Motivation

- Related works involve *Token Graphs*
  - A *k-token graph* $F_k(G)$ of a graph G is the graph whose
    - *vertices* are size-k vertex subsets of G
    - *edges* are defined under Token Sliding
  - One of the earliest results is [Alavi, Behzad, Erdős, and Lick 1991]
    - They studied the regularity, bipartitedness, Eulerianity, etc. of $F_2(G)$—which they called the *double vertex graphs*
  - *Combinatorial and algebraic properties* of $F_k(G)$ have been *extensively studied*, especially since it was *reintroduced in [Fabila-Monroy, Flores-Peñaloza, et al. 2012]*, e.g., see [Lew 2023]; [Fabila-Monroy and Trujillo-Negrete 2023]
- General Framework: *Reconfiguration Problems*
  - A *growing research area* in computer science which primarily aims to *understand the solution space of a computational problem*
  - Under the "reconfiguration setting", $\mathsf{TS}_k(G)$ has been *extremely well-studied* from the *algorithmic perspective* [van den Heuvel 2013]; [Nishimura 2018]; [Bousquet, Mouawad, Nishimura, and Siebertz 2022]
- We *initiate* the study of $\mathsf{TS}_k(G)$ from a purely *graph-theoretic perspective* [Avis and Hoang 2023a]; [Avis and Hoang 2023b]

Paths Are TS_k-Graphs

Paths Are TS_k-Graphs

- We say that a graph H is a TS_k*-graph* if there exists a graph G such that $H \simeq \mathsf{TS}_k(G)$
  - For example, C_5 is a TS_2-graph, since $C_5 \simeq \mathsf{TS}_2(C_5)$
- Since $G \simeq \mathsf{TS}_1(G)$ for any graph G, *the case $k = 1$ is trivial and uninteresting*. From now on, we will always consider *fixed integers $k \geq 2$*

Paths Are TS_k-Graphs

- We say that a graph H is a TS_k*-graph* if there exists a graph G such that $H \simeq \mathsf{TS}_k(G)$
  - For example, C_5 is a TS_2-graph, since $C_5 \simeq \mathsf{TS}_2(C_5)$
- Since $G \simeq \mathsf{TS}_1(G)$ for any graph G, *the case $k = 1$ is trivial and uninteresting*. From now on, we will always consider *fixed integers $k \geq 2$*

Theorem 1

A path P_n $(n \geq 1)$ is a TS_k-graph for any $k \geq 2$

- We briefly explain *three different ways* of proving Theorem 1, each of which *indicates a way of looking at TS_k-graphs*

Line Graphs and TS_2-Graphs

Lemma 2

Let $\overline{G}$ and $L(G)$ be respectively the complement and line graph of G

(a) $\mathsf{TS}_2(\overline{G})$ *is a (spanning) subgraph of* $L(G)$

(b) $\mathsf{TS}_2(\overline{G}) \simeq L(G)$ *if and only if G is triangle-free*

edges of G ⟷ non-edges of $\overline{G}$

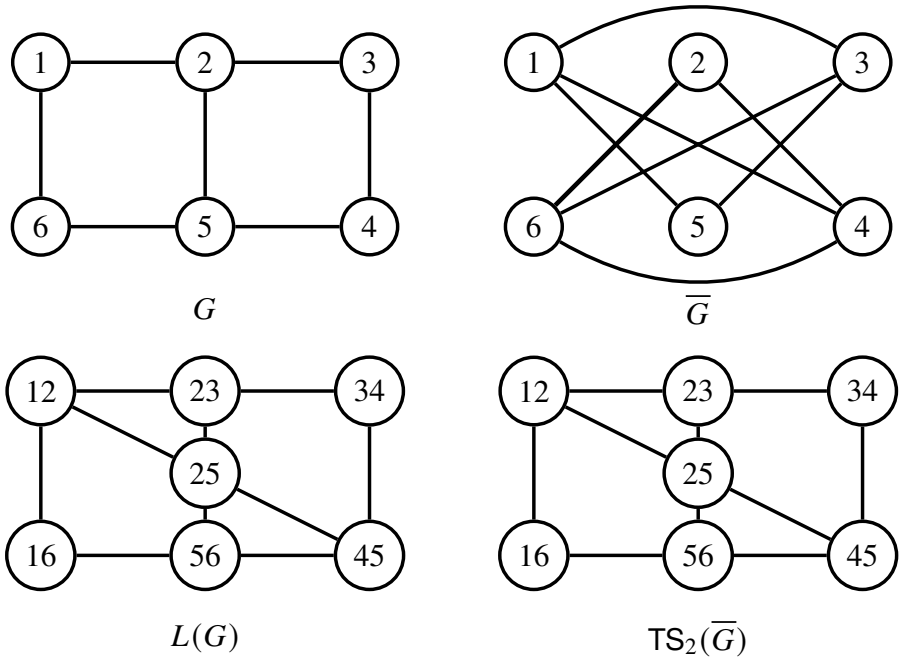
⇕ ⇕

vertices of $L(G)$ ⟷ vertices of $\mathsf{TS}_2(\overline{G})$

edges of a triangle of G ⟷ three size-2 stable sets of $\overline{G}$

⇕ ⇕

vertices of a triangle of $L(G)$ ⟷ a size-3 stable set of $\mathsf{TS}_2(\overline{G})$



Line Graphs and TS_2-Graphs

Lemma 2

Let $\overline{G}$ and $L(G)$ be respectively the complement and line graph of G

(a) $\mathsf{TS}_2(\overline{G})$ *is a (spanning) subgraph of* $L(G)$

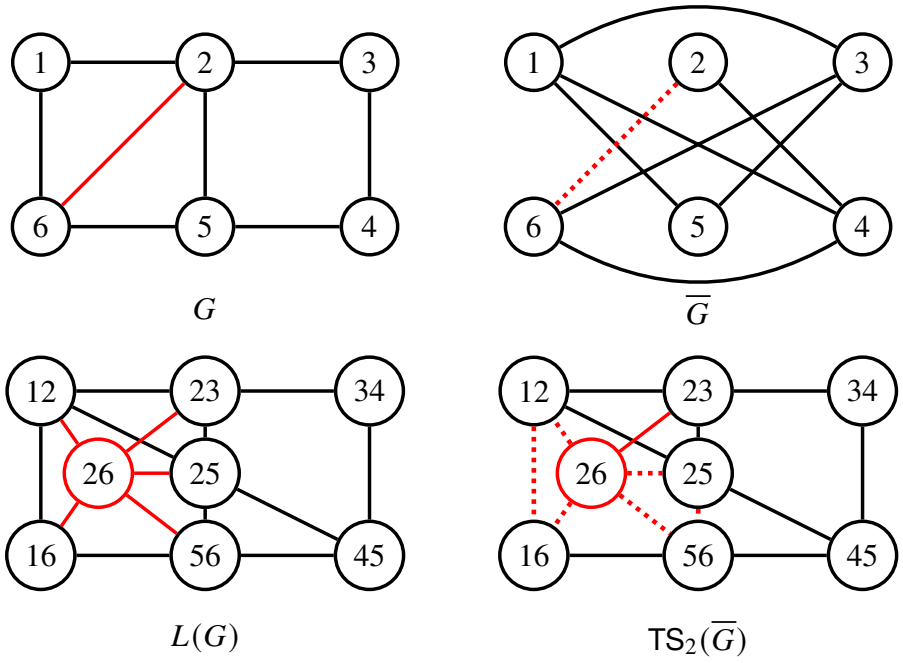
(b) $\mathsf{TS}_2(\overline{G}) \simeq L(G)$ *if and only if G is triangle-free*

edges of G ⟷ non-edges of $\overline{G}$

vertices of $L(G)$ ⟷ vertices of $\mathsf{TS}_2(\overline{G})$

edges of a triangle of G ⟷ three size-2 stable sets of $\overline{G}$

vertices of a triangle of $L(G)$ ⟷ a size-3 stable set of $\mathsf{TS}_2(\overline{G})$



Line Graphs and TS_2-Graphs

- Since P_n *is triangle-free* for any $n > 1$ and $P_n \simeq L(P_{n+1})$, Lemma 2 immediately gives us

$$P_n \simeq L(P_{n+1}) \simeq \mathsf{TS}_2(\overline{P_{n+1}})$$

Thus, P_n *is always a* TS_2*-graph*

Line Graphs and TS_2-Graphs

- Since *P_n is triangle-free* for any $n > 1$ and $P_n \simeq L(P_{n+1})$, Lemma 2 immediately gives us

$$P_n \simeq L(P_{n+1}) \simeq \mathsf{TS}_2(\overline{P_{n+1}})$$

Thus, *P_n is always a TS_2-graph*

- The following proposition *settles the case $k \geq 3$* (note that $\alpha(\overline{P_{n+1}}) = 2$)

Proposition 3

Given H and let $G = \mathsf{TS}_{\alpha(H)}(H)$. Then, for every $k \geq \alpha(H)$, G is a TS_k-graph

- Construct a graph H' by taking the *disjoint union of H and exactly $k - \alpha(H)$ isolated vertices*
- Then, $G = \mathsf{TS}_{\alpha(H)}(H) \simeq \mathsf{TS}_k(H')$ for any fixed integer $k \geq \alpha(H)$

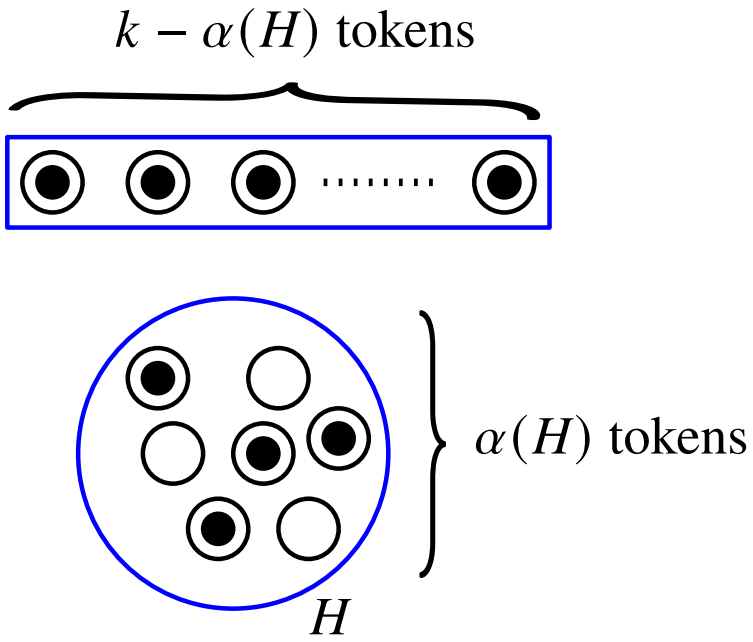


Figure: The graph H'

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

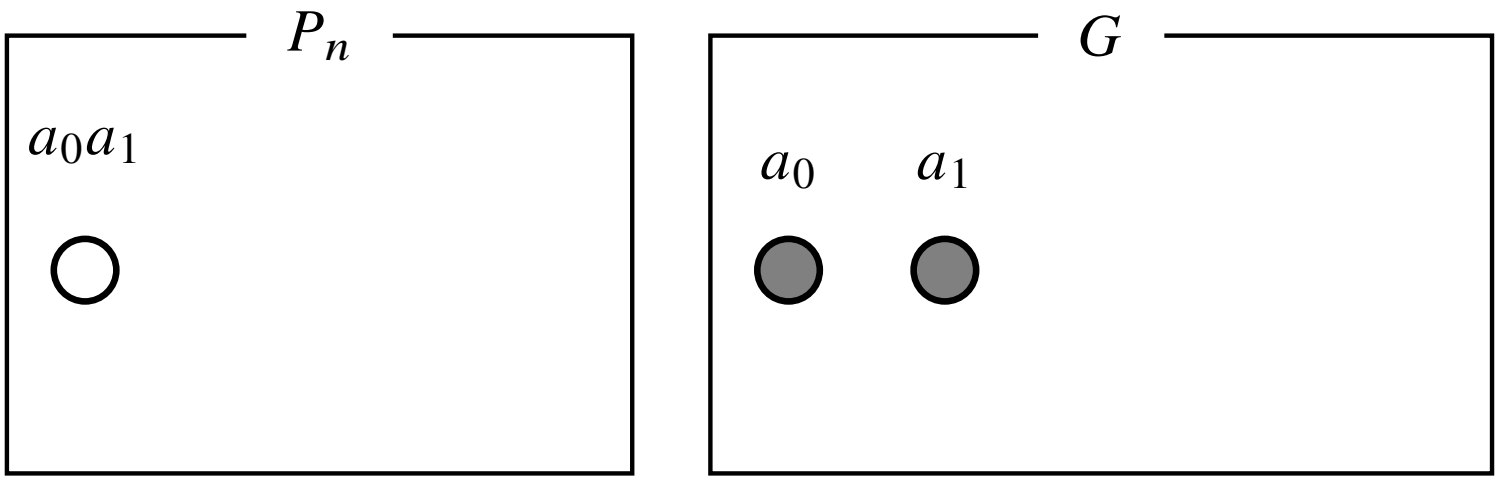


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

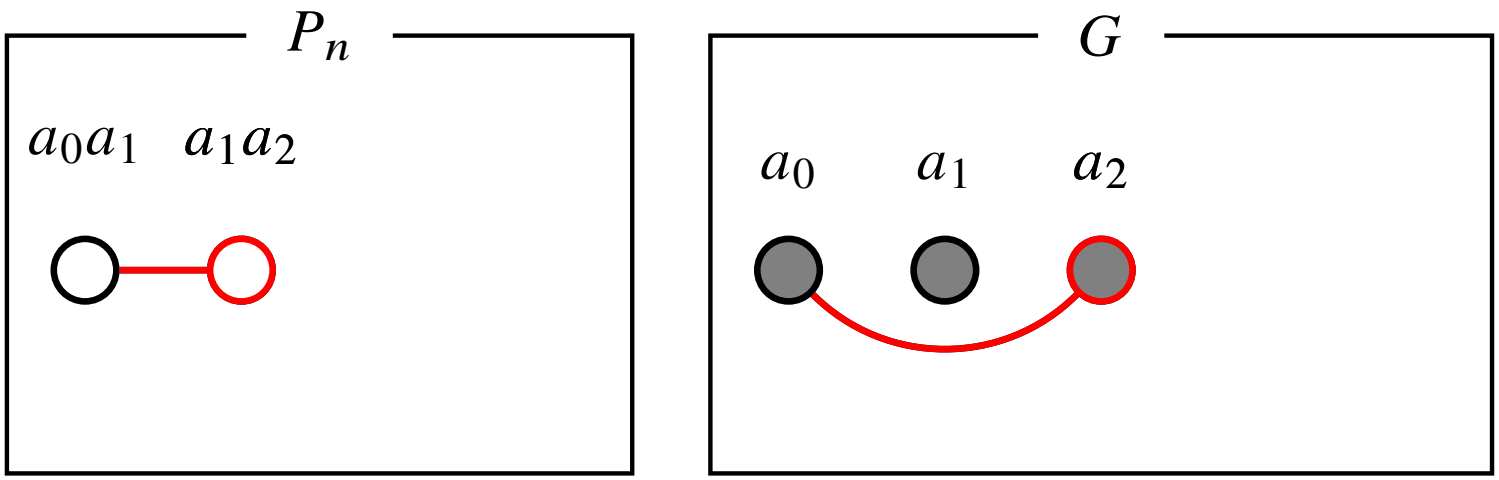


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

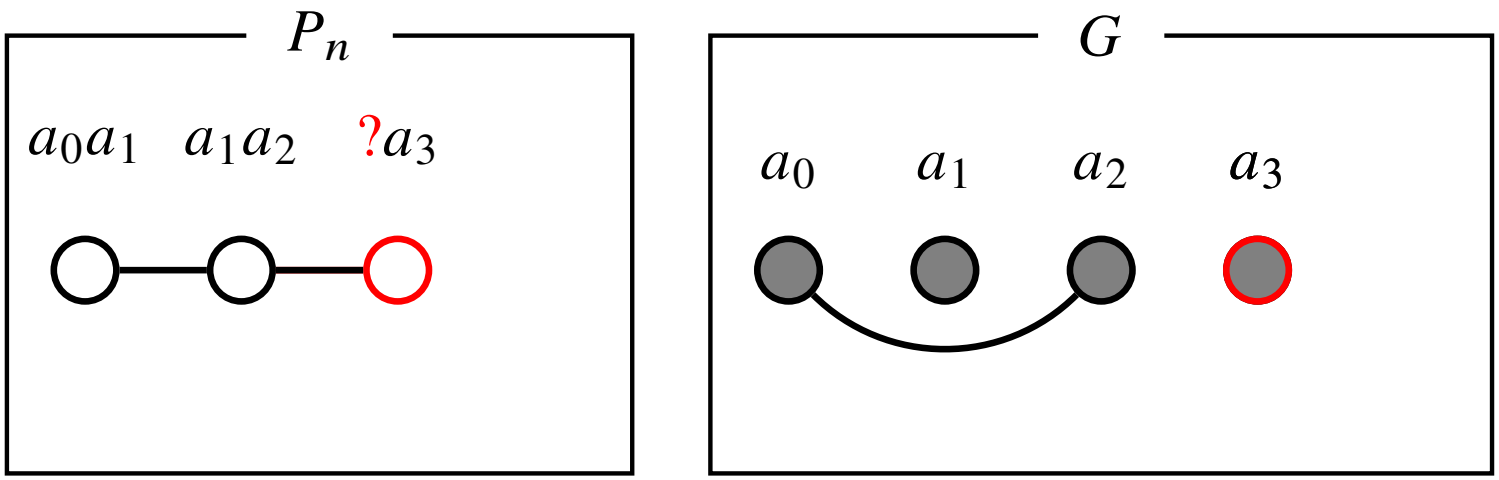


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

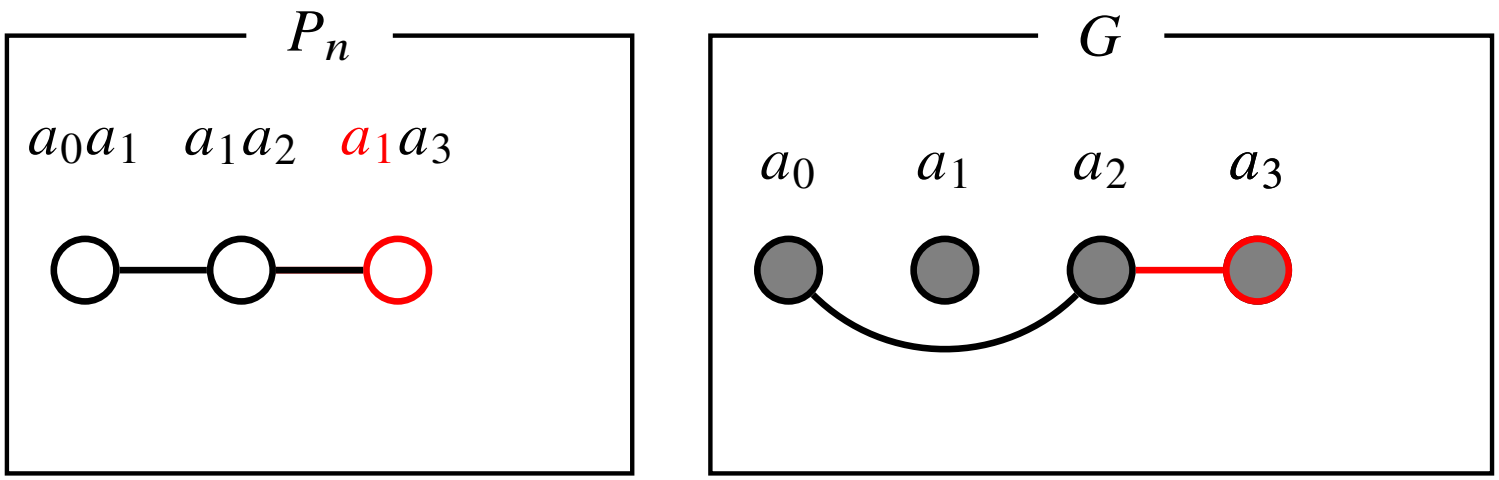


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

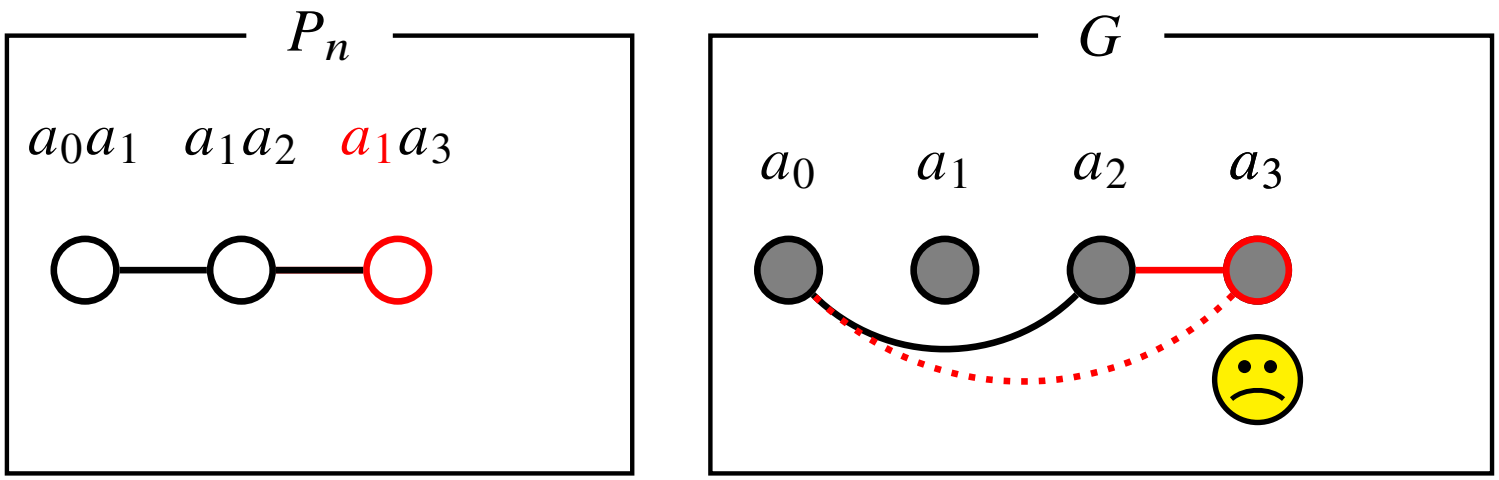


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

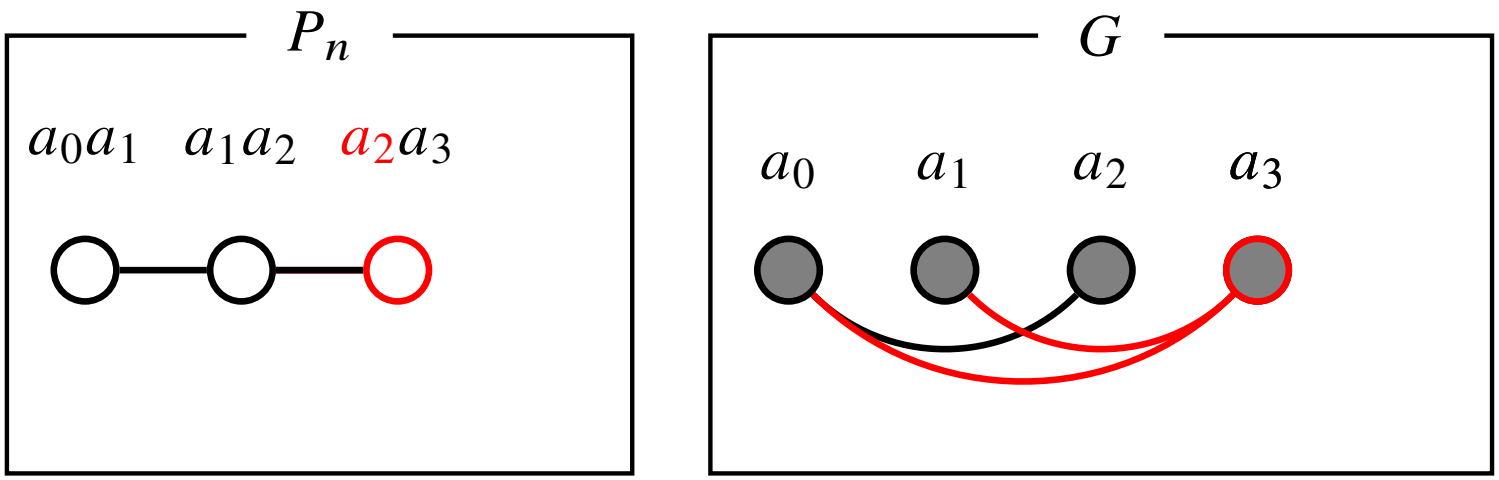


Figure: $P_n \simeq \mathsf{TS}_2(G)$

Labelling Vertices By Stable Sets

Key Idea ($k = 2$)

There exists G such that $P_1 \simeq \mathsf{TS}_2(G)$. For $n \geq 2$, we *update G and label vertices of P_n by size-2 stable sets of G*

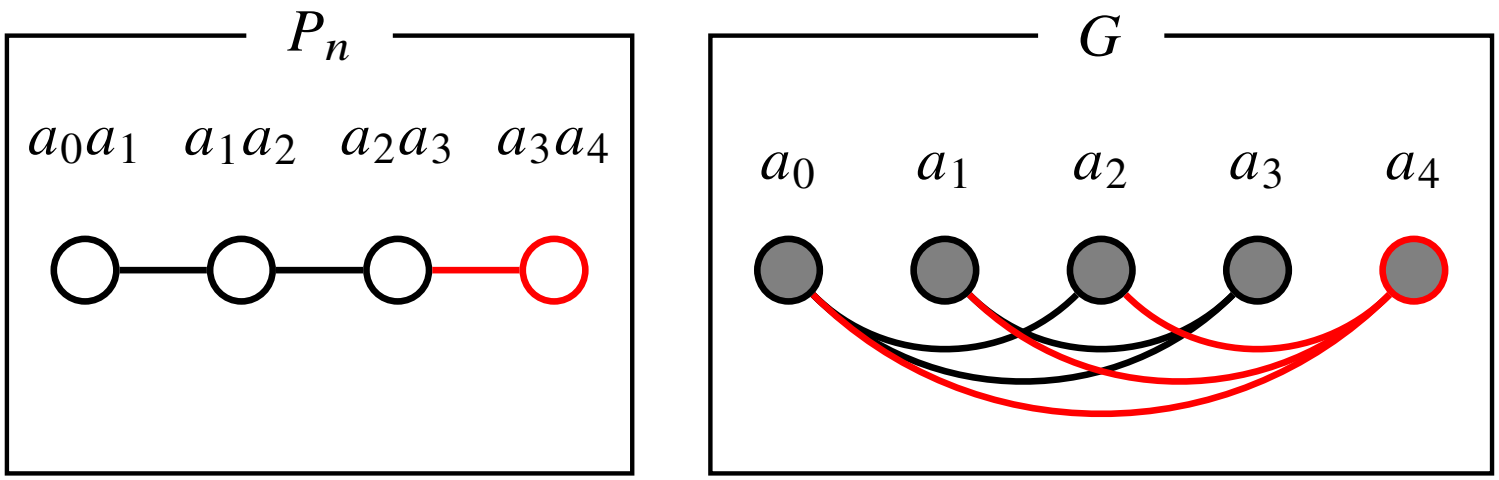


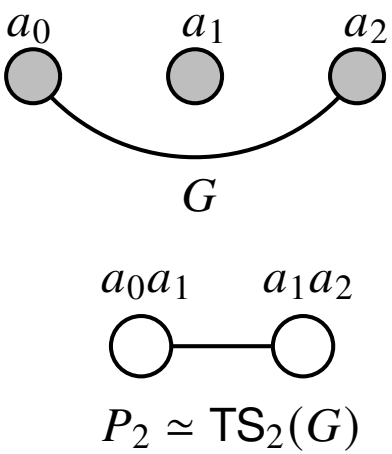
Figure: $P_n \simeq \mathsf{TS}_2(G)$

Gluing Labelled Graphs Together

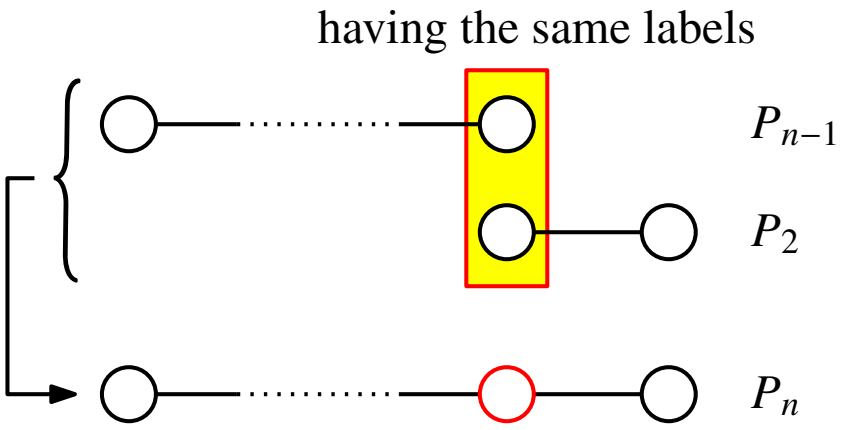
Key Idea ($k = 2$)

(a) There exists a (labelled) graph G such that $P_2 \simeq \mathsf{TS}_2(G)$

(b) With a "good" vertex-labelling (by stable sets of some graph), the TS_2-graph P_n can be constructed by *"gluing" two smaller* TS_2*-graphs* P_{n-1} *and* P_2, for any $n \geq 3$



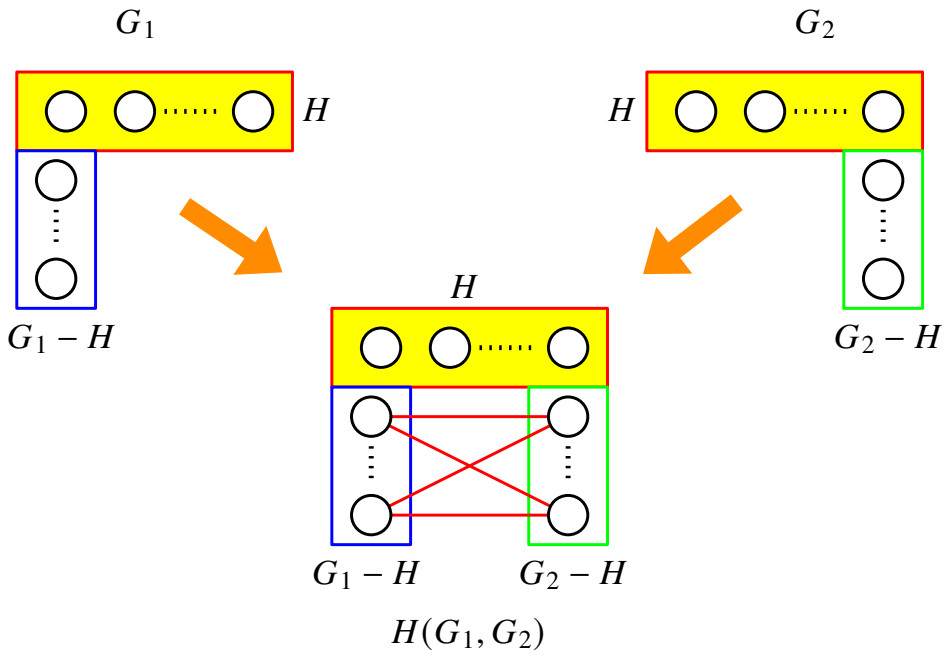
(a)



(b)

Gluing Labelled Graphs Together

- Vertex-labelled graphs G_1 and G_2 are *H-consistent* if *the (possibly empty) intersection of their vertex sets defines the same (possibly empty) common induced subgraph H*
- The *H-join of H-consistent graphs G_1, G_2*, denoted by $H(G_1, G_2)$, is the graph constructed by
  - Putting G_2 on top of G_1 (or vice versa) so that their common induced subgraph H coincide
  - Joining any pair of vertices v, w between $G_1 - H$ and $G_2 - H$, respectively



Gluing Labelled Graphs Together

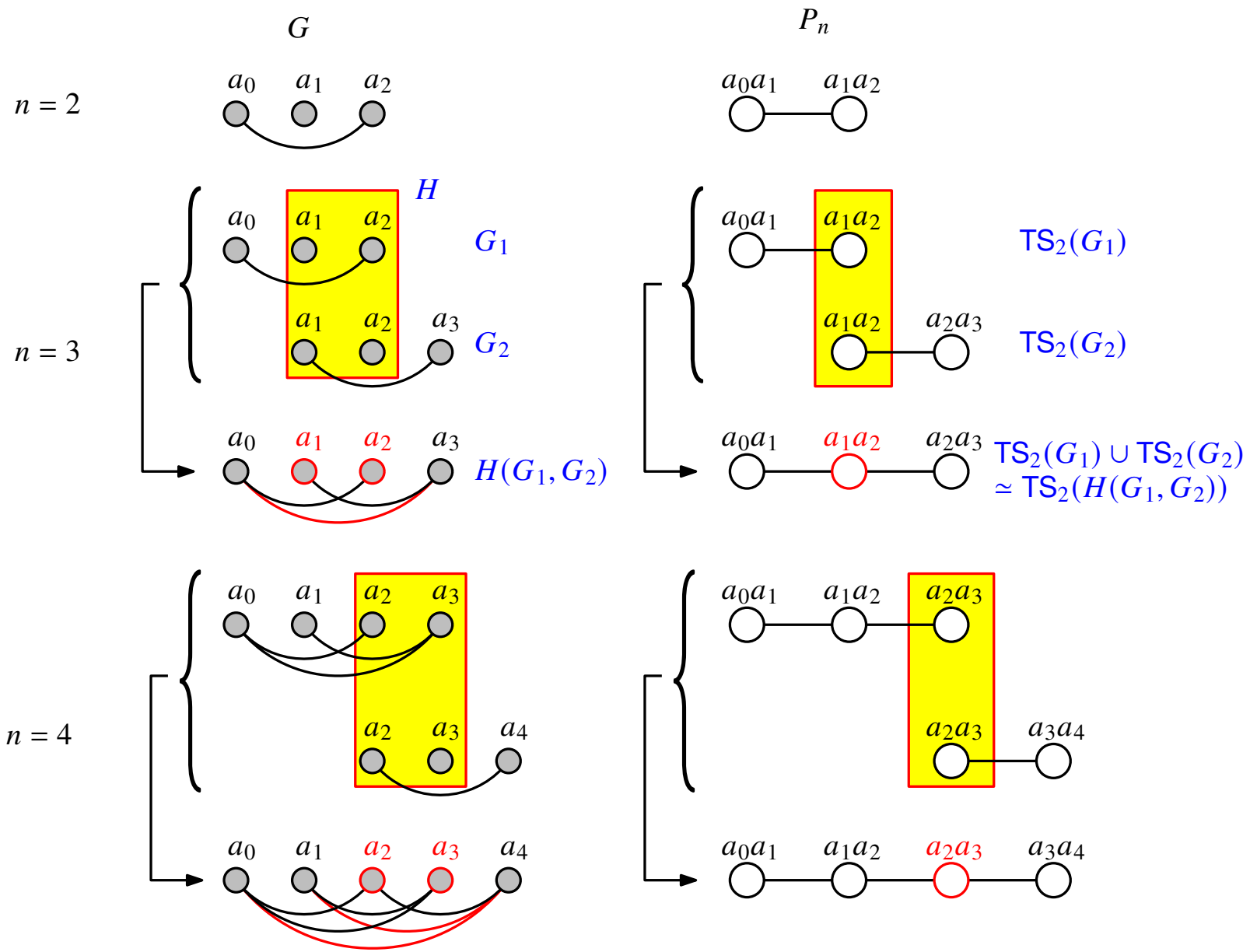


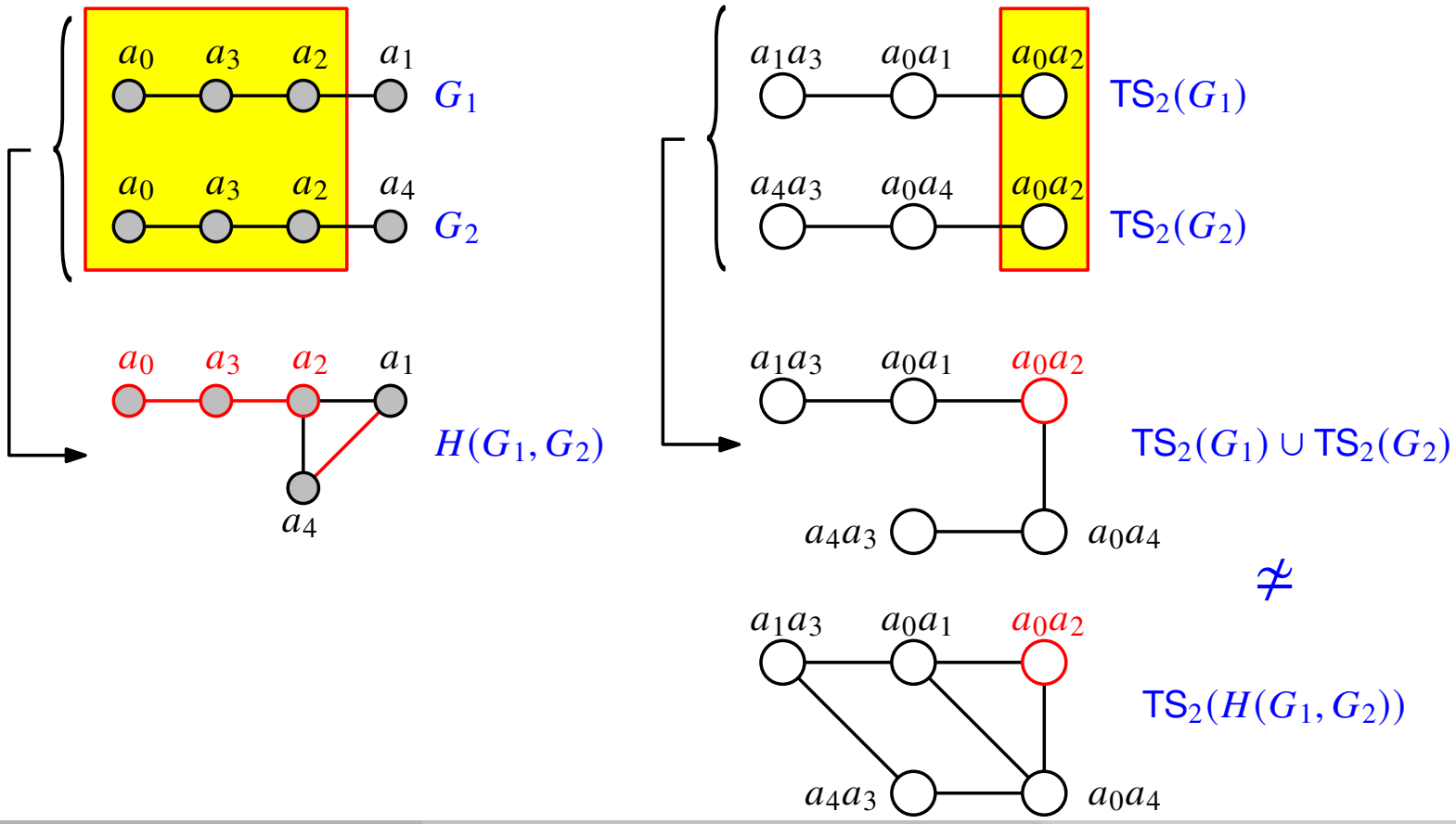
Figure: $P_n \simeq \mathsf{TS}_2(G)$

Gluing Labelled Graphs Together

Remark

There exist H-consistent graphs G_1, G_2 such that

$$\mathsf{TS}_2(G_1) \cup \mathsf{TS}_2(G_2) \not\approx \mathsf{TS}_2(H(G_1, G_2))$$



Gluing Labelled Graphs Together

In general, the following proposition describes *how to compute the* TS_k*-graph of an* H*-join*

Proposition 4

Let $k \geq 2$ *and let* G_1 *and* G_2 *be two* H*-consistent graphs.* $\mathsf{TS}_k(H(G_1, G_2))$ *is the union of* $\mathsf{TS}_k(G_1)$, $\mathsf{TS}_k(G_2)$ *and for every pair of* k*-element stable sets* S_1 *in* G_1 *and* S_2 *in* G_2 *satisfying*

$$|S_1 \cap V(H)| = |S_2 \cap V(H)| = |S_1 \cap S_2| = k - 1, \tag{1}$$

the edge between S_1 *and* S_2

Gluing Labelled Graphs Together

In general, the following proposition describes *how to compute the* TS_k*-graph of an* H*-join*

Proposition 4

Let $k \geq 2$ *and let* G_1 *and* G_2 *be two* H*-consistent graphs.* $\mathsf{TS}_k(H(G_1, G_2))$ *is the union of* $\mathsf{TS}_k(G_1)$, $\mathsf{TS}_k(G_2)$ *and for every pair of* k*-element stable sets* S_1 *in* G_1 *and* S_2 *in* G_2 *satisfying*

$$|S_1 \cap V(H)| = |S_2 \cap V(H)| = |S_1 \cap S_2| = k - 1, \tag{1}$$

the edge between S_1 *and* S_2

- We say that $H(G_1, G_2)$ is *k-crossing free* if there are *no k-element stable sets satisfying condition (1) of Proposition 4*

Corollary 5

Let $k \geq 2$ *and let* G_1 *and* G_2 *be two* H*-consistent graphs.* $H(G_1, G_2)$ *is* k*-crossing free if and only if*

$$\mathsf{TS}_k(H(G_1, G_2)) \simeq \mathsf{TS}_k(G_1) \cup \mathsf{TS}_k(G_2) \tag{2}$$

On TS_k-Graphs

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$) for any $k \geq 2$

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$) for any $k \geq 2$
- K_n $(n \geq 1)$ is a TS_k-graph. (A direct consequence of Proposition 3)

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$) for any $k \geq 2$
- *K_n $(n \geq 1)$* is a TS_k-graph. (A direct consequence of Proposition 3)
- *P_n $(n \geq 1)$ and C_n $(n \geq 3)$* are TS_k-graphs for any $k \geq 2$. (Use the "line graph" approach)

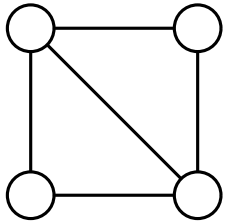
On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$) for any $k \geq 2$
- *K_n $(n \geq 1)$* is a TS_k-graph. (A direct consequence of Proposition 3)
- *P_n $(n \geq 1)$ and C_n $(n \geq 3)$* are TS_k-graphs for any $k \geq 2$. (Use the "line graph" approach)
- *$K_{m,n}$ $(1 \leq m \leq n)$* is a TS_k-graph for any $k \geq 2$ if and only if either $m = 1$ and $n \leq k$ or $m = n = 2$ (Use the "labelling vertices by stable sets" approach. A similar approach can be used to characterize whether *split graphs* are (not) TS_k-graphs)

On TS_k-Graphs

- If H is an induced subgraph of G then $\mathsf{TS}_k(H)$ is also an induced subgraph of $\mathsf{TS}_k(G)$. The reverse does not hold (e.g., take $H = C_{2k}$ and $G = K_{1,k+1}$) for any $k \geq 2$
- *K_n $(n \geq 1)$* is a TS_k-graph. (A direct consequence of Proposition 3)
- *P_n $(n \geq 1)$ and C_n $(n \geq 3)$* are TS_k-graphs for any $k \geq 2$. (Use the "line graph" approach)
- *$K_{m,n}$ $(1 \leq m \leq n)$* is a TS_k-graph for any $k \geq 2$ if and only if either $m = 1$ and $n \leq k$ or $m = n = 2$ (Use the "labelling vertices by stable sets" approach. A similar approach can be used to characterize whether *split graphs* are (not) TS_k-graphs)

- For any $k \geq 2$,
  - *K_1 is the smallest graph that is a TS_k-graph*
  - On the other hand, the *diamond* is the *smallest graph that is not a TS_k-graph*

G	H such that $G \simeq \mathsf{TS}_k(H)$ $(k \geq 2)$
 K_1	 k isolated vertices
 diamond	does not exist

On Acyclic TS_k-Graphs

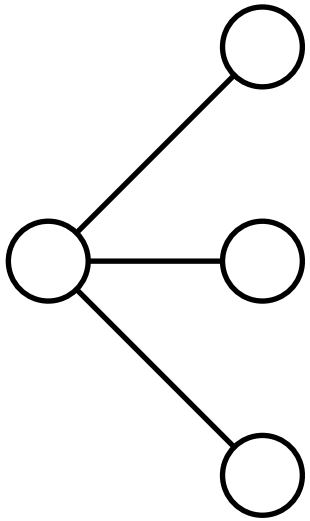
On Acyclic TS_k-Graphs

Remark

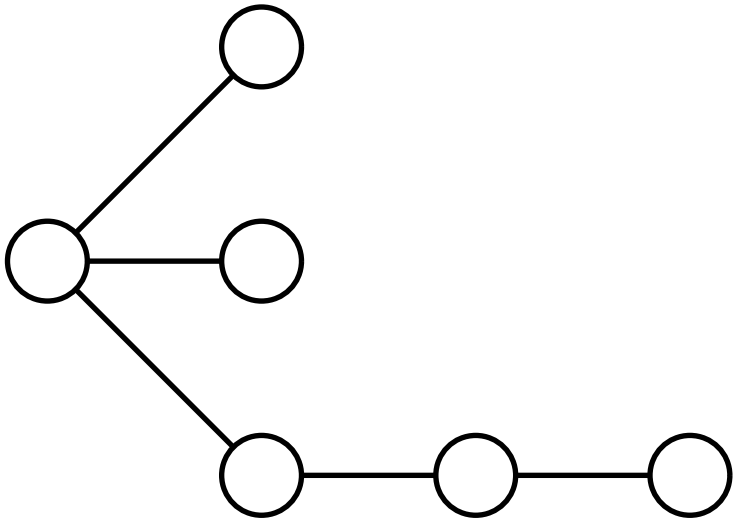
"Being a TS_k-graph" is *not* hereditary, even when $k = 2$

Using the "labelling vertices by stable sets" approach, we showed that

- $K_{1,3}$ is *not* a TS_2-graph. More generally, *$K_{1,n}$ is a TS_k-graph for some fixed $k \geq 2$ if and only if $n \leq k$*
- Replacing an edge of $K_{1,3}$ by a P_4 results a TS_2-graph



not a TS_2-graph



a TS_2-graph

On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

(2) Under which conditions trees are (not) TS_k-graphs?

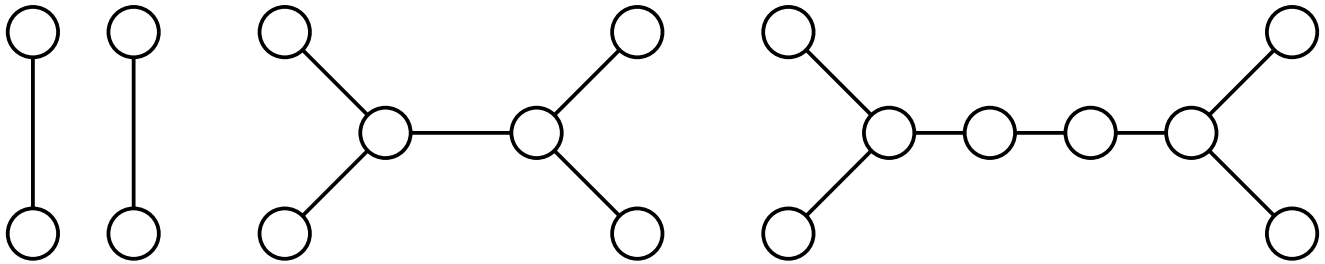
On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

(2) Under which conditions trees are (not) TS_k-graphs?

- Given a forest F
  - $\mathsf{TS}_2(F)$ is acyclic if and only if F is $\{2K_2, D_{2,2,2}\}$-free



$2K_2$ $D_{2,2,2}$ $D_{2,4,2}$

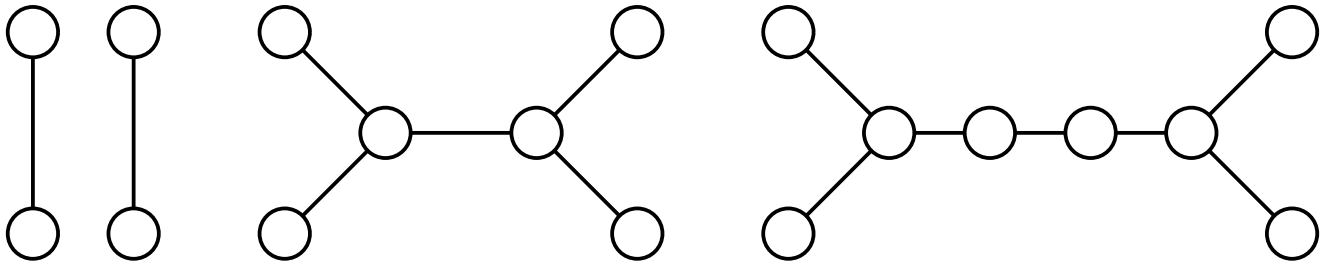
On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

(2) Under which conditions trees are (not) TS_k-graphs?

- Given a forest F
  - $\mathsf{TS}_2(F)$ is acyclic if and only if F is $\{2K_2, D_{2,2,2}\}$-free
  - $\mathsf{TS}_3(F)$ is acyclic if and only if F is $\{2K_2 + K_1, D_{2,2,2} + K_1, D_{2,4,2}\}$-free



$2K_2$ $\quad$ $D_{2,2,2}$ $\quad$ $D_{2,4,2}$

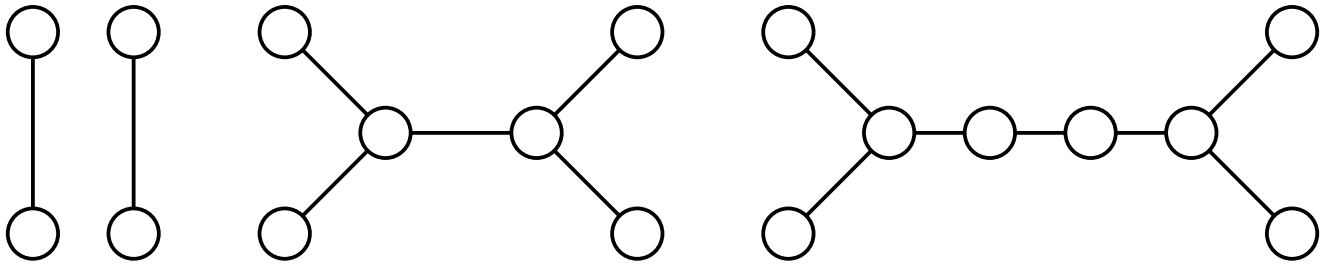
On Acyclic TS_k-Graphs

Open Question

(1) Under which conditions a given graph G satisfies $\mathsf{TS}_k(G)$ is acyclic?

(2) Under which conditions trees are (not) TS_k-graphs?

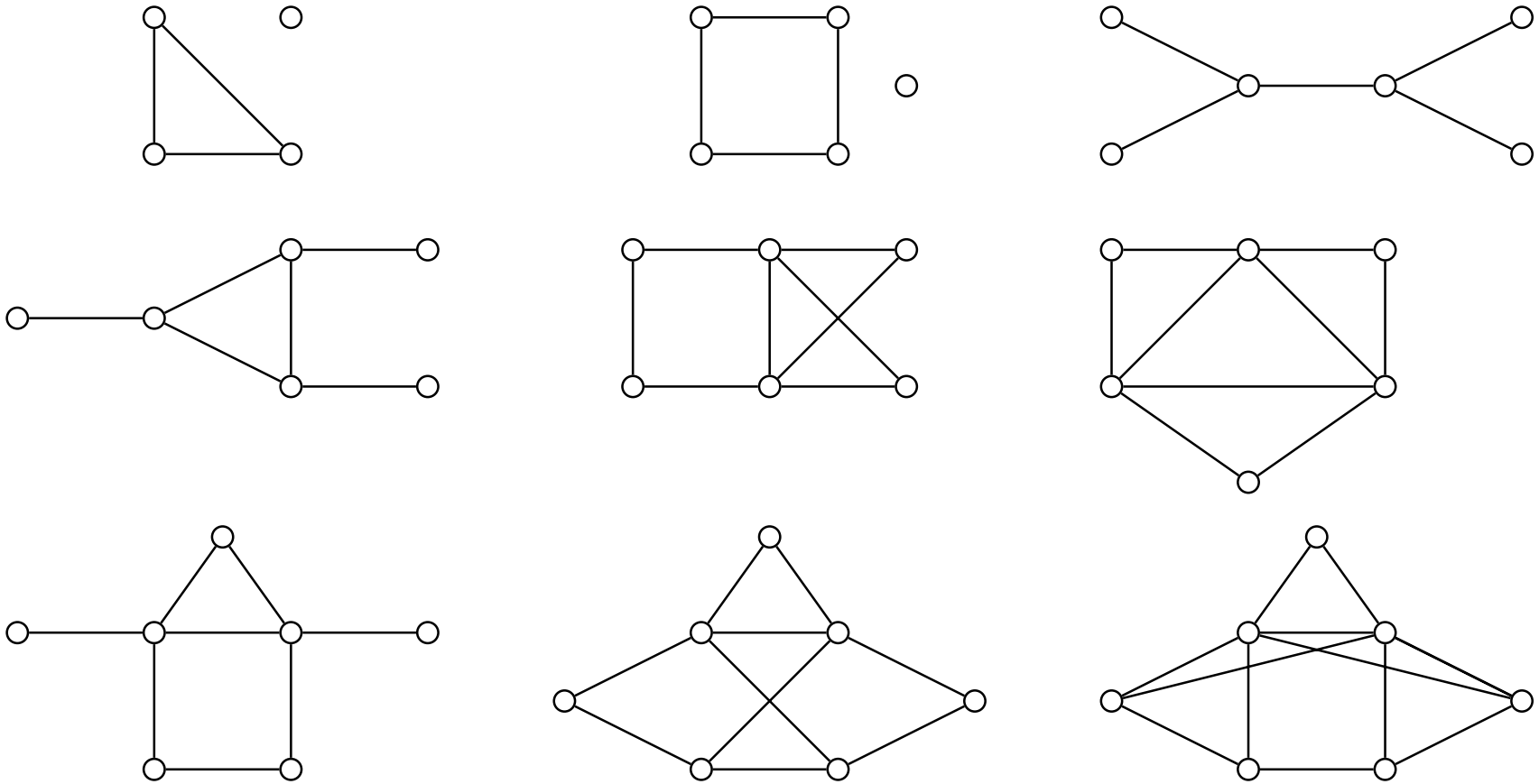
- Given a forest F
  - $\mathsf{TS}_2(F)$ is acyclic if and only if F is $\{2K_2, D_{2,2,2}\}$-free
  - $\mathsf{TS}_3(F)$ is acyclic if and only if F is $\{2K_2 + K_1, D_{2,2,2} + K_1, D_{2,4,2}\}$-free
  - **Conjecture:** $\mathsf{TS}_k(F)$ is acyclic if and only if F is $\{2K_2 + (k-2)K_1, D_{2,2,2} + (k-2)K_1, D_{2,4,2} + (k-3)K_1\}$-free, for $k \geq 4$



$2K_2$ $\quad$ $D_{2,2,2}$ $\quad$ $D_{2,4,2}$

On Acyclic TS_k-Graphs

- Given a graph G. If G contains either $\overline{C_n}$ *($n \geq 4$)* or *one of the following nine graphs* as an induced subgraph then $\mathsf{TS}_2(G)$ has a cycle
  - The $\overline{C_n}$ ($n \geq 4$) graphs come from the "line graph" approach (Lemma 2)
  - (Most of) The below nine graphs come from a computer program



- **Open Question:** Does the reverse hold? (i.e., Did we miss any graph?)

On Acyclic TS_k-Graphs

Using the "gluing graphs together" approach, we showed that

- A *n-ary tree* is a rooted tree in which each node has at most n children. (A 2-ary tree is the well-known binary tree.) *For $k \geq 2$, any k-ary tree is a TS_{k+1}-graph*

On Acyclic TS_k-Graphs

Using the "gluing graphs together" approach, we showed that

- A *n-ary tree* is a rooted tree in which each node has at most n children. (A 2-ary tree is the well-known binary tree.) *For $k \geq 2$, any k-ary tree is a TS_{k+1}-graph*
- For every tree T, there exists a graph G such that $\mathsf{TS}_2(G) \simeq T + \ell K_1$ *for some integer ℓ*. Thus, for every tree T, there exists a graph G such that $\mathsf{TS}_2(G)$ *is a forest containing T*
- **Open Question:** For $k \geq 3$ and a tree T, is there a graph G such that $\mathsf{TS}_k(G)$ is a forest containing T?

On Acyclic TS_k-Graphs

Using the "gluing graphs together" approach, we showed that

- A *n-ary tree* is a rooted tree in which each node has at most n children. (A 2-ary tree is the well-known binary tree.) *For $k \geq 2$, any k-ary tree is a TS_{k+1}-graph*
- For every tree T, there exists a graph G such that $\mathsf{TS}_2(G) \simeq T + \ell K_1$ *for some integer ℓ*. Thus, for every tree T, there exists a graph G such that $\mathsf{TS}_2(G)$ *is a forest containing T*
- **Open Question:** For $k \geq 3$ and a tree T, is there a graph G such that $\mathsf{TS}_k(G)$ is a forest containing T?
- For every tree T, there exists a graph G such that $\mathsf{TS}_2(G)$ *is a tree containing T* if and only if T is a 3-ary tree
- **Open Question:** For $k \geq 3$ and a $(k+1)$-ary tree T, is there a graph G such that $\mathsf{TS}_k(G)$ is a tree containing T?

On Acyclic TS_k-Graphs

Using the "gluing graphs together" approach, we showed that

- A *n-ary tree* is a rooted tree in which each node has at most n children. (A 2-ary tree is the well-known binary tree.) *For $k \geq 2$, any k-ary tree is a TS_{k+1}-graph*
- For every tree T, there exists a graph G such that $\mathsf{TS}_2(G) \simeq T + \ell K_1$ *for some integer ℓ*. Thus, for every tree T, there exists a graph G such that $\mathsf{TS}_2(G)$ *is a forest containing T*
- **Open Question:** For $k \geq 3$ and a tree T, is there a graph G such that $\mathsf{TS}_k(G)$ is a forest containing T?
- For every tree T, there exists a graph G such that $\mathsf{TS}_2(G)$ *is a tree containing T* if and only if T is a 3-ary tree
- **Open Question:** For $k \geq 3$ and a $(k+1)$-ary tree T, is there a graph G such that $\mathsf{TS}_k(G)$ is a tree containing T?
- There exists a graph G such that $\mathsf{TS}_k(G)$ *is a tree containing $K_{1,n}$* if $n \leq 2k$
- **Open Question:** Does there exist a graph G such that $\mathsf{TS}_k(G)$ contains $K_{1,n}$ for $n > 2k$?

On Acyclic TS_k-Graphs

For graphs in the *$D_{r,n,s}$ family ($1 \leq r \leq s$)*,

- *For $n = 1$*, $D_{r,n,s}$ is nothing but the star $K_{1,r+s}$ and therefore it is a TS_k-graph if and only if $r + s \leq k$

On Acyclic TS_k-Graphs

For graphs in the *$D_{r,n,s}$ family ($1 \leq r \leq s$)*,

- *For $n = 1$*, $D_{r,n,s}$ is nothing but the star $K_{1,r+s}$ and therefore it is a TS_k-graph if and only if $r + s \leq k$
- *For $n \geq 2$ and $k \geq 2$*, if $s \leq k - 1$, $D_{r,n,s}$ is always a TS_k-graph (in this case, it is a $(k-1)$-ary tree)

On Acyclic TS_k-Graphs

For graphs in the *$D_{r,n,s}$ family ($1 \leq r \leq s$)*,

- *For $n = 1$*, $D_{r,n,s}$ is nothing but the star $K_{1,r+s}$ and therefore it is a TS_k-graph if and only if $r + s \leq k$
- *For $n \geq 2$ and $k \geq 2$*, if $s \leq k - 1$, $D_{r,n,s}$ is always a TS_k-graph (in this case, it is a $(k-1)$-ary tree)
  - *The reverse does not hold in general*: for example, take $r = 1$, $s = k = 2$, and $n = 3$

On Acyclic TS_k-Graphs

For graphs in the *$D_{r,n,s}$ family ($1 \leq r \leq s$)*,

- *For $n = 1$*, $D_{r,n,s}$ is nothing but the star $K_{1,r+s}$ and therefore it is a TS_k-graph if and only if $r + s \leq k$
- *For $n \geq 2$ and $k \geq 2$*, if $s \leq k - 1$, $D_{r,n,s}$ is always a TS_k-graph (in this case, it is a $(k-1)$-ary tree)
  - *The reverse does not hold in general*: for example, take $r = 1$, $s = k = 2$, and $n = 3$
  - Indeed, *$D_{1,n,2}$ is a TS_2-graph if and only if $n = 3$*. Thus, $D_{1,3,2}$ is the only TS_2-graph among all $D_{1,n,2}$ for $n \geq 1$

On Acyclic TS_k-Graphs

For graphs in the *$D_{r,n,s}$ family ($1 \leq r \leq s$)*,

- *For $n = 1$*, $D_{r,n,s}$ is nothing but the star $K_{1,r+s}$ and therefore it is a TS_k-graph if and only if $r + s \leq k$
- *For $n \geq 2$ and $k \geq 2$*, if $s \leq k - 1$, $D_{r,n,s}$ is always a TS_k-graph (in this case, it is a $(k-1)$-ary tree)
  - *The reverse does not hold in general*: for example, take $r = 1$, $s = k = 2$, and $n = 3$
  - Indeed, *$D_{1,n,2}$ is a TS_2-graph if and only if $n = 3$*. Thus, $D_{1,3,2}$ is the only TS_2-graph among all $D_{1,n,2}$ for $n \geq 1$
  - However, *the reverse does hold for $n = 2$*, that is, $D_{r,2,s}$ is a TS_k-graph if and only if $s \leq k - 1$

References

- Avis, D. and D. A. Hoang (2023a). "A Note On Acyclic Token Sliding Reconfiguration Graphs of Independent Sets". In: *arXiv preprint*. arXiv: 2301.00317.
- Avis, D. and D. A. Hoang (2023b). "On Reconfiguration Graph of Independent Sets under Token Sliding". In: *Graphs and Combinatorics* 39.3. (article 59). DOI: 10.1007/s00373-023-02644-w. arXiv: 2203.16861.
- Fabila-Monroy, R. and A. L. Trujillo-Negrete (2023). "On the Automorphism Group of Token Graphs of Complete Bipartite Graphs and Cartesian Products of Connected Graphs". In: *arXiv preprint*. arXiv: 2302.07914.
- Lew, A. (2023). "Garland's method for token graphs". In: *arXiv preprint*. arXiv: 2305.02406.

References

- Bousquet, N., A. E. Mouawad, N. Nishimura, and S. Siebertz (2022). "A survey on the parameterized complexity of the independent set and (connected) dominating set reconfiguration problems". In: *arXiv preprint*. arXiv: 2204.10526.
- Nishimura, N. (2018). "Introduction to reconfiguration". In: *Algorithms* 11.4. (article 52). DOI: 10.3390/a11040052.
- van den Heuvel, J. (2013). "The complexity of change". In: *Surveys in Combinatorics*. Vol. 409. London Math. Soc. Lecture Note Ser. Cambridge University Press, pp. 127–160. DOI: 10.1017/CBO9781139506748.005.
- Fabila-Monroy, R., D. Flores-Peñaloza, C. Huemer, F. Hurtado, J. Urrutia, and D. R. Wood (2012). "Token Graphs". In: *Graphs and Combinatorics* 28.3, pp. 365–380. DOI: 10.1007/s00373-011-1055-9.

References

Alavi, Y., M. Behzad, P. Erdős, and D. R. Lick (1991). “Double Vertex Graphs”. In: *Journal of Combinatorics, Information & System Sciences* 16.1, pp. 37–50.