Token sliding on graphs of girth five

Demian Banakh
Department of Theoretical Computer Science
Jagiellonian University

January 26, 2023

Notation

- $[n] = \{1, \ldots, n\}$
- All graphs are finite, simple and undirected
- *Open neighborhood* $N_G(v) = \{u \mid uv \in E(G)\}$
- *Closed neighborhood* $N_G[v] = N_G(v) \cup v$
- For $W \subseteq V(G)$, let $N_G(Q) = \bigcup_Q N_G(v) - Q$ and $N_G[Q] = N_G(Q) \cup Q$
- *Diameter* of G is $\mathrm{diam}(G) = \max_{v,u} \mathrm{dist}_G(v, u)$
- *Girth* of G is the length of the shortest cycle in G

Token sliding problem

Input: graph G and 2 independent k-sets $I_s, I_t \subseteq G$.
Question: whether there is a sequence of independent k-sets $(I_0, \ldots, I_l)$ such that

$$I_0 = I_s, I_l = I_t,$$
$$I_i \Delta I_{i+1} = \{u, v\} \in E(G)$$

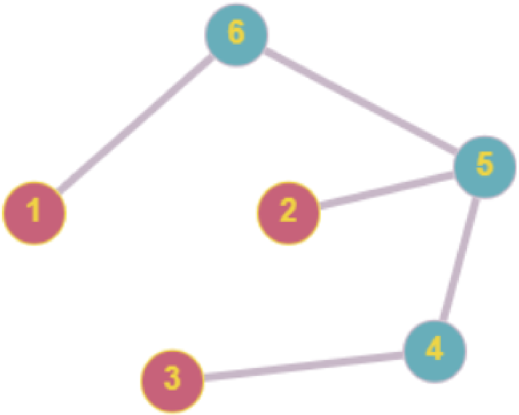
Token sliding problem

Input: graph G and 2 independent k-sets $I_s, I_t \subseteq G$.
Question: whether there is a sequence of independent k-sets $(I_0, \ldots, I_l)$ such that

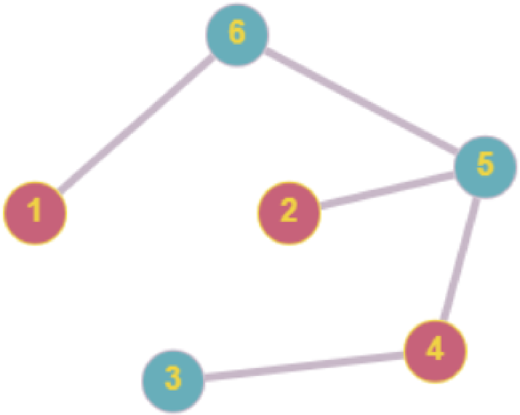
$$I_0 = I_s, I_l = I_t,$$
$$I_i \Delta I_{i+1} = \{u, v\} \in E(G)$$

If we call vertices of I_i *tokens*, then every *move* from I_i to I_{i+1} is "sliding" one token along the edge maintaining the independence.

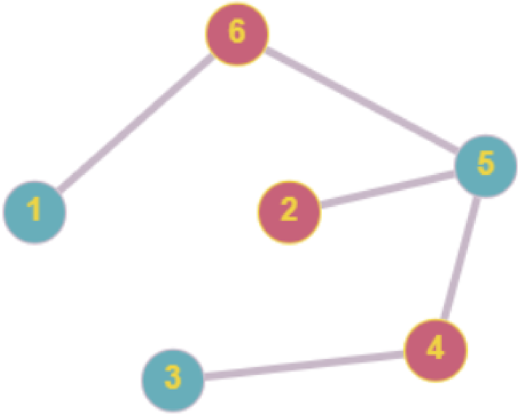
Token sliding problem



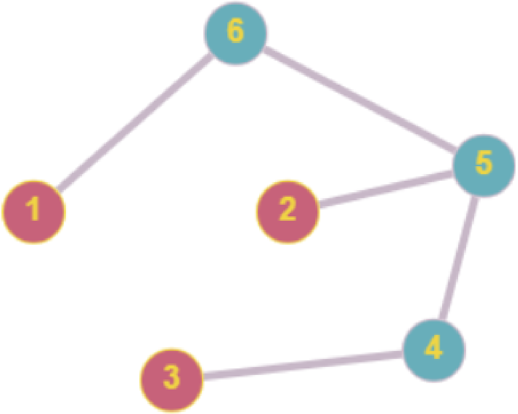
Token sliding problem



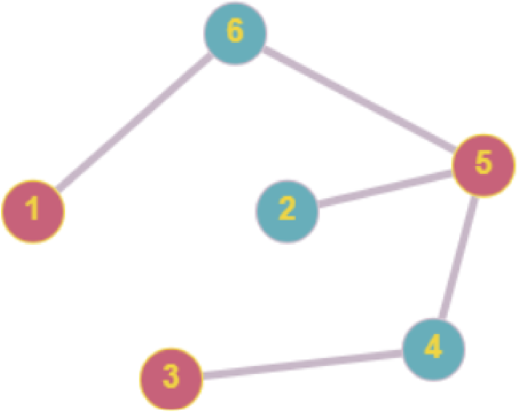
Token sliding problem



Token sliding problem



Token sliding problem



Token sliding problem

- TOKEN SLIDING is **PSPACE**-complete

Token sliding problem

- TOKEN SLIDING is **PSPACE**-complete
- TOKEN SLIDING is in **P** on trees, interval graphs etc.

Token sliding problem

- TOKEN SLIDING is **PSPACE**-complete
- TOKEN SLIDING is in **P** on trees, interval graphs etc.
- In 2021, the same authors proved that TOKEN SLIDING is **W[1]**-hard for graphs of girth 4 or less

Token sliding problem

- TOKEN SLIDING is **PSPACE**-complete
- TOKEN SLIDING is in **P** on trees, interval graphs etc.
- In 2021, the same authors proved that TOKEN SLIDING is **W[1]**-hard for graphs of girth 4 or less

TOKEN SLIDING can be solved naively by constructing a *reconfiguration graph* $\mathcal{R}(G, k)$, where vertices are independent k-sets of G, and edges correspond to *moves*. Then it's enough to verify if I_t is reachable from I_s - $O(n^k)$.

Main result

Fixed-parameter tractable - $O(f(k) \cdot n^{O(1)})$.

Main result

Fixed-parameter tractable - $O(f(k) \cdot n^{O(1)})$.

Theorem

TOKEN SLIDING is fixed-parameter tractable when parameterized by k on graphs of girth ≥ 5.

Main result

Fixed-parameter tractable - $O(f(k) \cdot n^{O(1)})$.

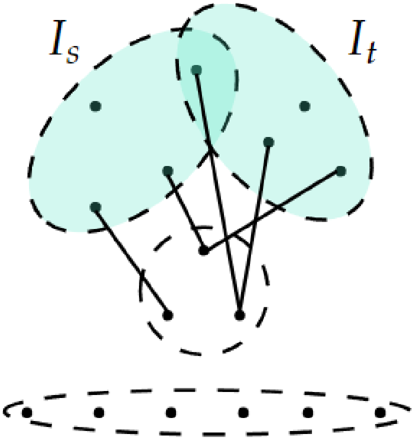
Theorem

TOKEN SLIDING is fixed-parameter tractable when parameterized by k on graphs of girth ≥ 5.

Goal: bound the size of G by $f(k)$, and apply the naive algorithm.

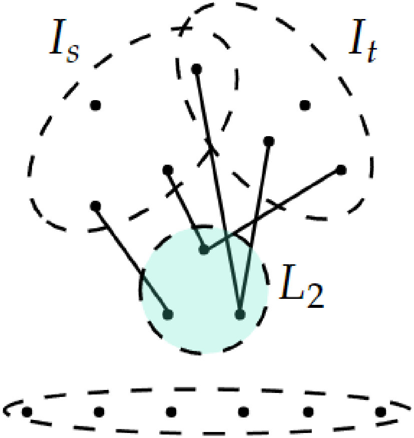
Partition G

- Let $L_1 = I_s \cup I_t$,



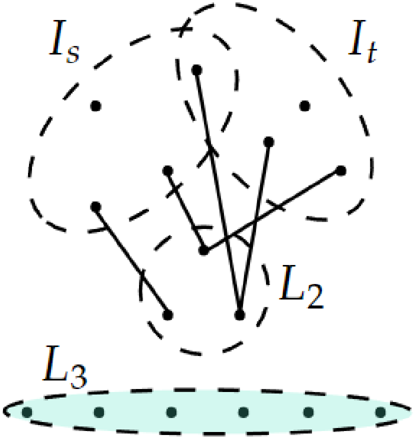
Partition G

- Let $L_1 = I_s \cup I_t$,
- $L_2 = N(L_1)$, and



Partition G

- Let $L_1 = I_s \cup I_t$,
- $L_2 = N(L_1)$, and
- $L_3 = V \setminus (L_1 \cup L_2)$



Partition G

Lemma

If $u \in L_2 \cup L_3$, then $|N_{L_1 \cup L_2}(u)| \leq 2k$.

4 component types

Let C be max connected component in $G[L_3]$.

Definition

C is diameter-safe if $diam(G[C]) > k^3$

Definition

C is degree-safe if $\exists u \in C.\ N_{G[C]}(u) > k^2$ and $|\{v \in N_{G[C]}(u) \mid \deg_{G[C]}(v) = 2\}| \geq k^2$

Definition

C is bounded if $diam(G[C]) \leq k^3$ and $\forall u \in C.\ \deg_{G[C]}(u) \leq k^2$

Definition

C is bad otherwise

Bounded components

Definition

C is bounded if $diam(G[C]) \leq k^3$ and $\forall u \in C.\ \deg_{G[C]}(u) \leq k^2$

Lemma

If C is a bounded component in $G[L_3]$, then $|V(C)| \leq k^{2k^3}$.

Safe components - informally

We will be trying to show that for a safe component C

- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$

Safe components - informally

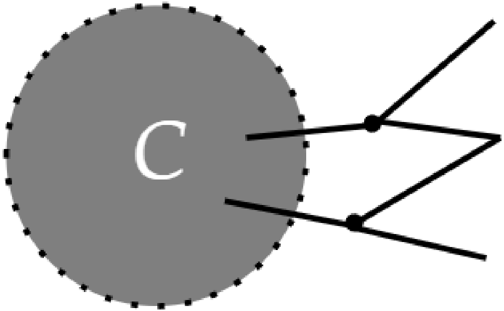
We will be trying to show that for a safe component C

- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed

Safe components - informally

We will be trying to show that for a safe component C

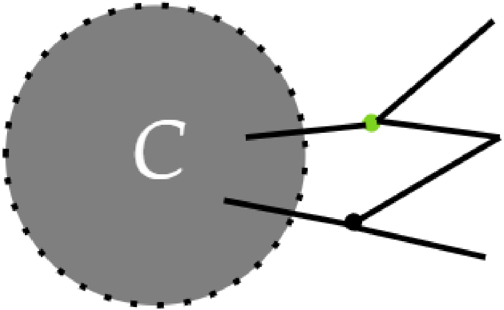
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Safe components - informally

We will be trying to show that for a safe component C

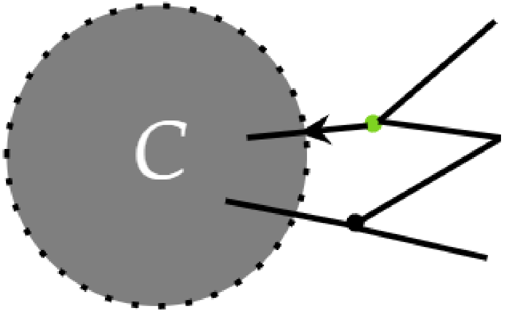
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Safe components - informally

We will be trying to show that for a safe component C

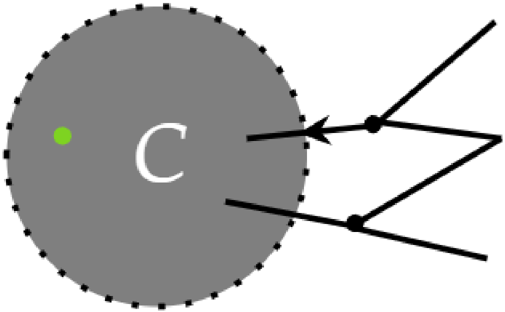
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Safe components - informally

We will be trying to show that for a safe component C

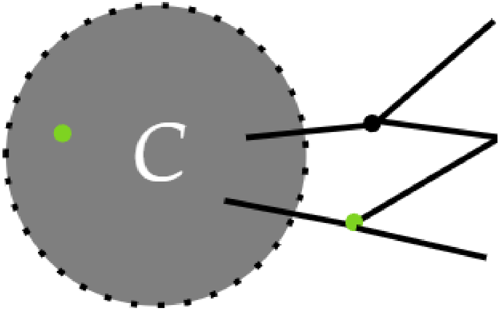
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Safe components - informally

We will be trying to show that for a safe component C

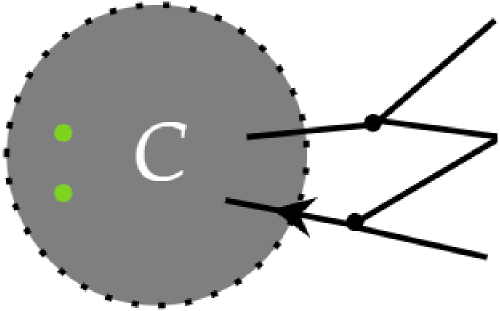
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Safe components - informally

We will be trying to show that for a safe component C

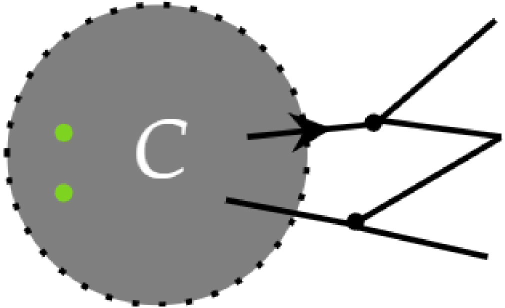
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Safe components - informally

We will be trying to show that for a safe component C

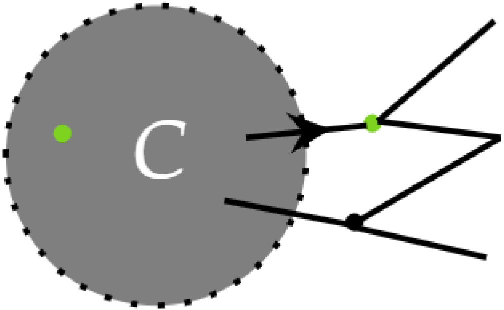
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Safe components - informally

We will be trying to show that for a safe component C

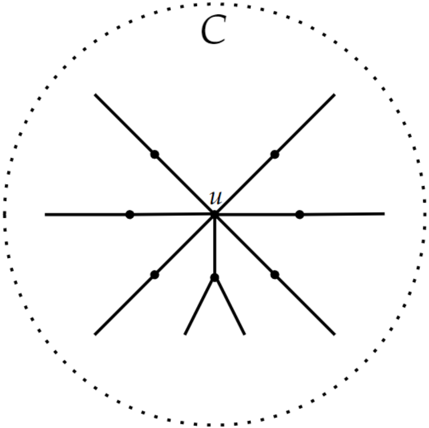
- if a sequence $\hat{I}$ from I_s to I_t exists, then also a sequence $\hat{I}'$ exists such that $|\hat{I}'_i \cap N_G(C)| \leq 1$
- it can absorb all k tokens, and project them back into C neighbourhood as needed



Degree-safe components

Definition

C is degree-safe if $\exists u \in C.\ N_{G[C]}(u) > k^2$ and $|\{v \in N_{G[C]}(u) \mid \deg_{G[C]}(v) = 2\}| \geq k^2$



Degree-safe components

Lemma

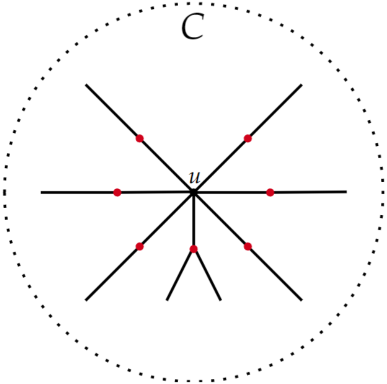
A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- Let $N_1(u) = N_C(u)$,

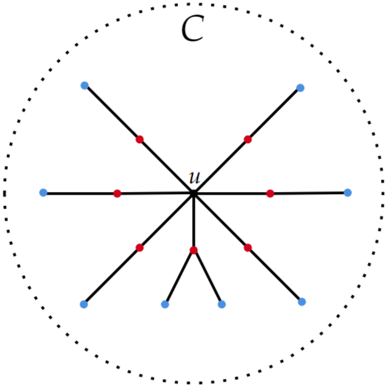


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- Let $N_1(u) = N_C(u)$,
- and $N_2(u) = N_C(N_1(u)) - u$

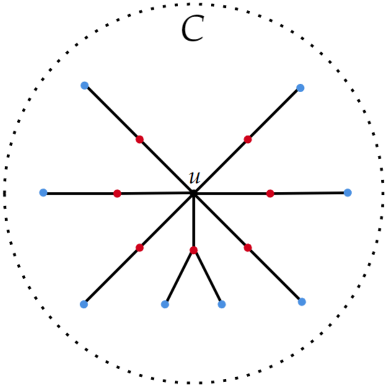


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- Let $N_1(u) = N_C(u)$,
- and $N_2(u) = N_C(N_1(u)) - u$
- $N_1(u)$ is an independent set, as G has no triangles

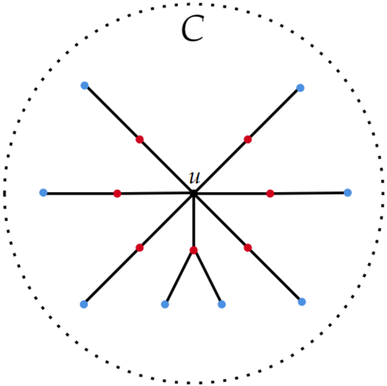


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- Let $N_1(u) = N_C(u)$,
- and $N_2(u) = N_C(N_1(u)) - u$
- $N_1(u)$ is an independent set, as G has no triangles
- $\forall v \in N_2(u).\ |N_{N_1(u)}(v)| \leq 1$, as G has no 4-cycles

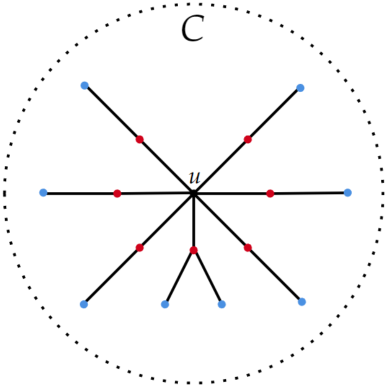


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- Let $N_1(u) = N_C(u)$,
- and $N_2(u) = N_C(N_1(u)) - u$
- $N_1(u)$ is an independent set, as G has no triangles
- $\forall v \in N_2(u).\ |N_{N_1(u)}(v)| \leq 1$, as G has no 4-cycles
- $|N_2(u)| \geq k^2$

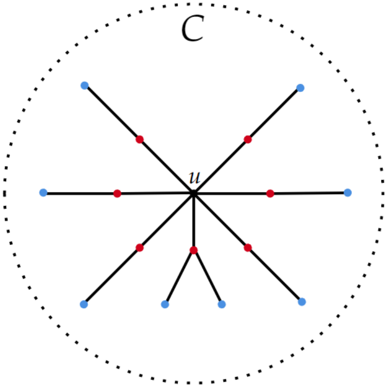


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- $|N_2(u)| \geq k^2$
- **Claim:** $G[N_2(u)]$ contains independent k-set. Why?

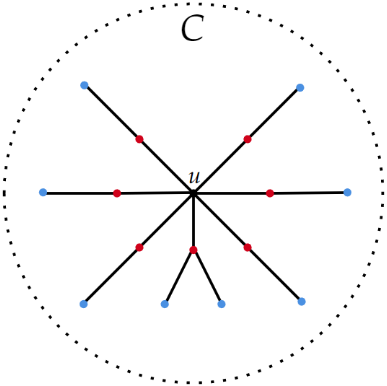


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- $|N_2(u)| \geq k^2$
- **Claim:** $G[N_2(u)]$ contains independent k-set. Why?
- Either $\exists v \in N_2(u)$ $\deg(v) \geq k$. Then $N(v)$ contains independent k-set, as G has no triangles

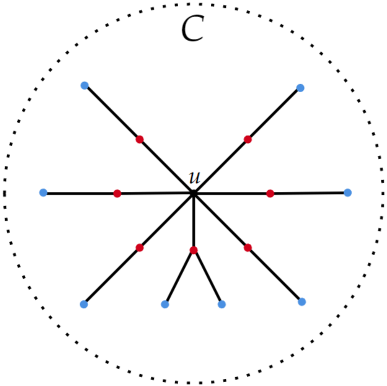


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

- $|N_2(u)| \geq k^2$
- **Claim:** $G[N_2(u)]$ contains independent k-set. Why?
- Either $\exists v \in N_2(u)$ $\deg(v) \geq k$. Then $N(v)$ contains independent k-set, as G has no triangles
- Or $\forall v \in N_2(u)$ $\deg(v) \leq k-1$. Then repeat k times: add any v to IS and remove $N[v]$ from $G[N_2(u)]$

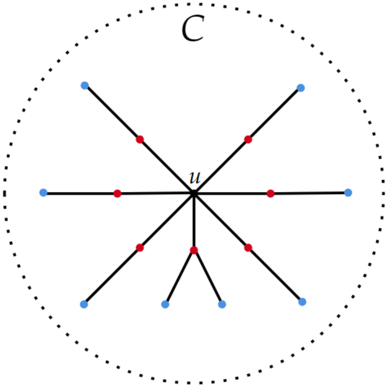


Degree-safe components

Lemma

A degree-safe component C in $G[L_3]$ contains an induced subdivided k-star, where all k branches have length > 1.

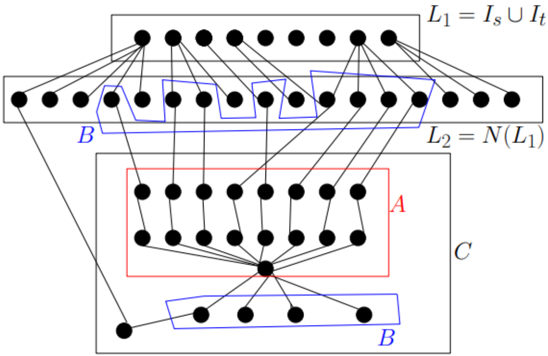
- $|N_2(u)| \geq k^2$
- **Claim:** $G[N_2(u)]$ contains independent k-set. Why?
- Either $\exists v \in N_2(u)$ $\deg(v) \geq k$. Then $N(v)$ contains independent k-set, as G has no triangles
- Or $\forall v \in N_2(u)$ $\deg(v) \leq k-1$. Then repeat k times: add any v to IS and remove $N[v]$ from $G[N_2(u)]$
- This claim gives us the desired subdivided k-star



Degree-safe components

Lemma

Let A be an induced subdivided k-star in C with all branches of length exactly 2. Let $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

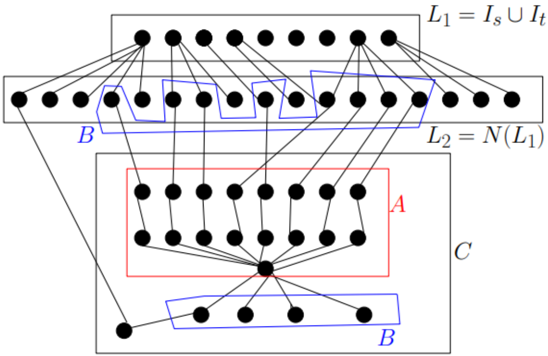


Degree-safe components

Lemma

Let A be an induced subdivided k-star in C with all branches of length exactly 2. Let $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$

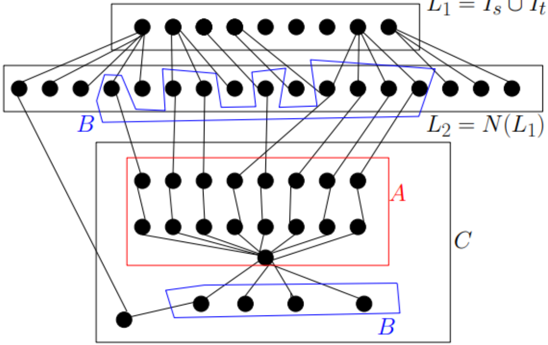


Degree-safe components

Lemma

Let A be an induced subdivided k-star in C with all branches of length exactly 2. Let $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$
- Let r be root of the star A,

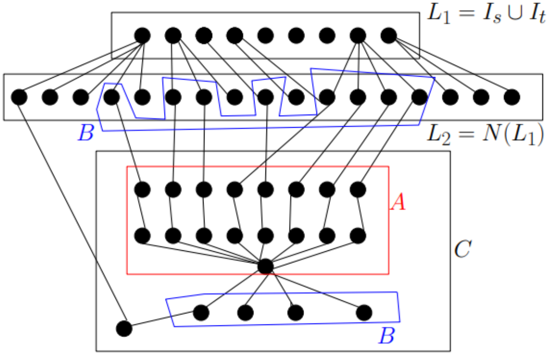


Degree-safe components

Lemma

Let A be an induced subdivided k-star in C with all branches of length exactly 2. Let $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$
- Let r be root of the star A,
- $N_1 = N_A(r)$ and $N_2 = N_A(N_1) - r$

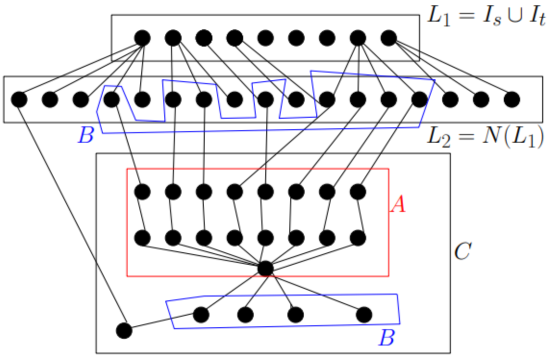


Degree-safe components

Lemma

Let A be an induced subdivided k-star in C with all branches of length exactly 2. Let $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$
- Let r be root of the star A,
- $N_1 = N_A(r)$ and $N_2 = N_A(N_1) - r$
- We will convert a sequence $\hat{I}$ from I_s to I_t into a sequence $\hat{I}'$ such that
  1. B never has > 1 tokens in $\hat{I}'_i$,

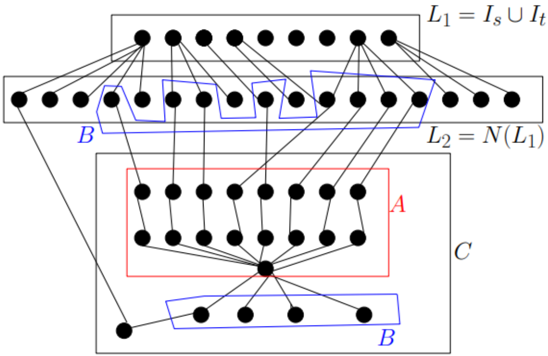


Degree-safe components

Lemma

Let A be an induced subdivided k-star in C with all branches of length exactly 2. Let $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$
- Let r be root of the star A,
- $N_1 = N_A(r)$ and $N_2 = N_A(N_1) - r$
- We will convert a sequence $\hat{I}$ from I_s to I_t into a sequence $\hat{I}'$ such that
  1. B never has > 1 tokens in $\hat{I}'_i$,
  2. at any step #tokens in $A \cup B$ is same and

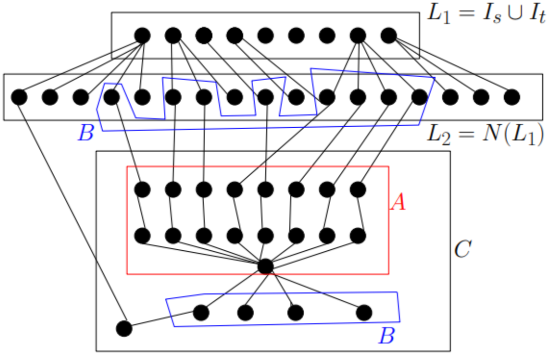


Degree-safe components

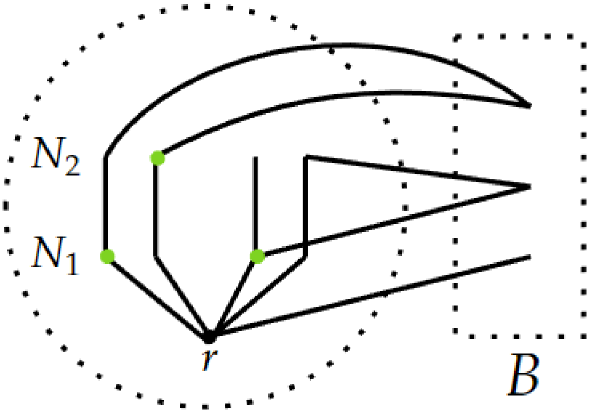
Lemma

Let A be an induced subdivided k-star in C with all branches of length exactly 2. Let $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$
- Let r be root of the star A,
- $N_1 = N_A(r)$ and $N_2 = N_A(N_1) - r$
- We will convert a sequence $\hat{I}$ from I_s to I_t into a sequence $\hat{I}'$ such that
  1. B never has > 1 tokens in $\hat{I}'_i$,
  2. at any step #tokens in $A \cup B$ is same and
  3. positions of tokens in $\overline{A \cup B}$ are same

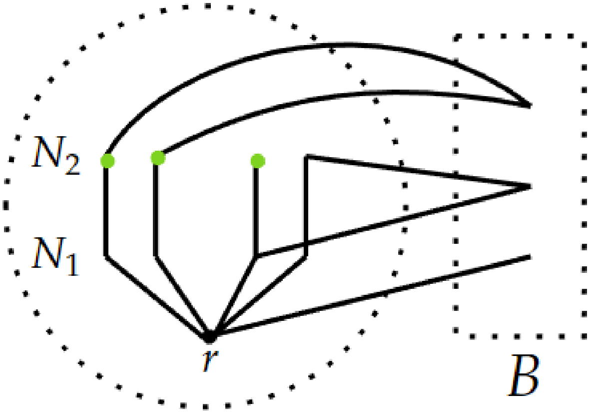


Degree-safe components



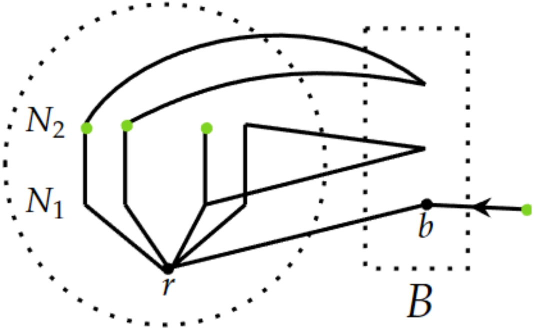
- Consider step I_{i-1}, right before some token enters vertex $b \in B$

Degree-safe components



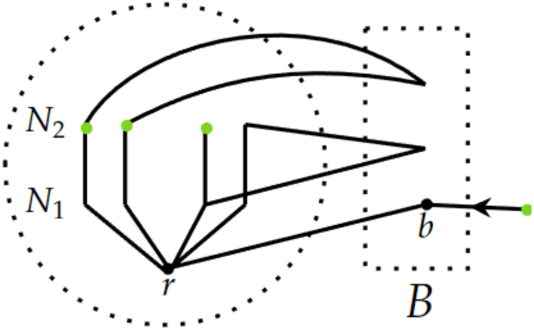
- Consider step I_{i-1}, right before some token enters vertex $b \in B$
- First, we move all tokens in A to N_2 - now every token has its own branch

Degree-safe components



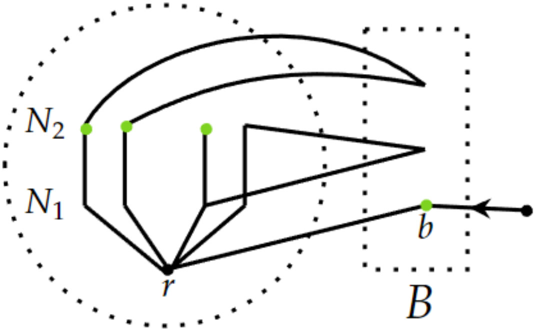
- **Case** $b \in N(r)$:

Degree-safe components



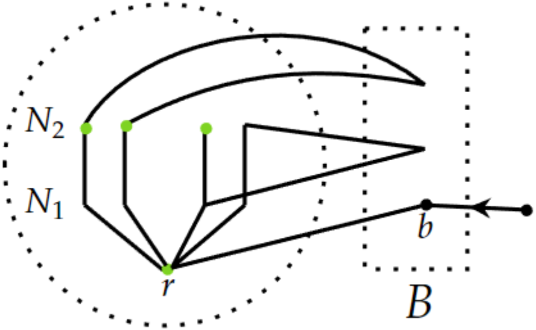
- **Case** $b \in N(r)$:
  - $b \notin N(N_2)$, as G has no 4-cycles

Degree-safe components



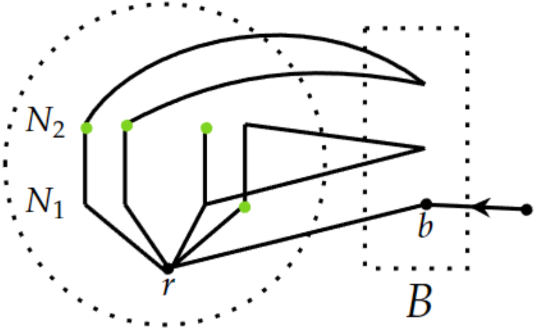
- **Case** $b \in N(r)$:
  - $b \notin N(N_2)$, as G has no 4-cycles
  - slide $b \to r \to$ any empty branch

Degree-safe components



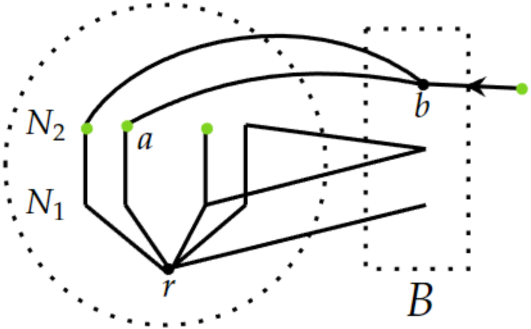
- **Case** $b \in N(r)$:
  - $b \notin N(N_2)$, as G has no 4-cycles
  - slide $b \to r \to$ any empty branch

Degree-safe components



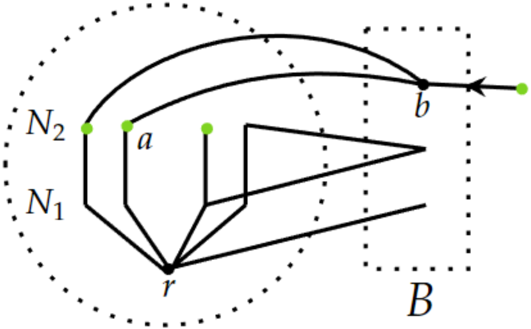
- **Case** $b \in N(r)$:
  - $b \notin N(N_2)$, as G has no 4-cycles
  - slide $b \to r \to$ any empty branch

Degree-safe components



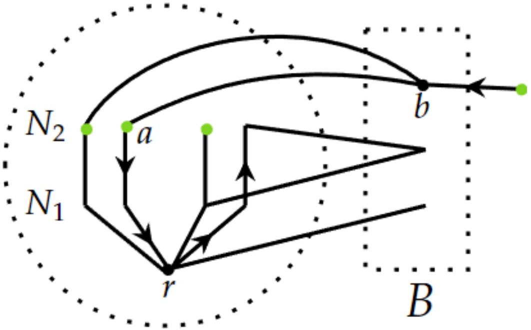
- **Case** $b \notin N(N_1)$:

Degree-safe components



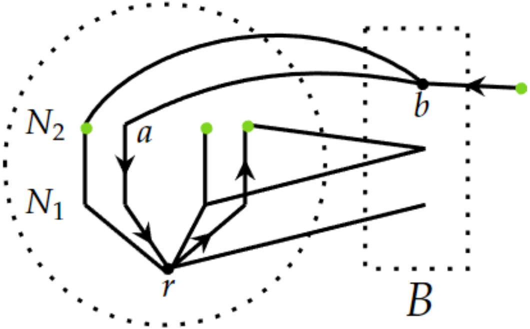
- **Case** $b \notin N(N_1)$:
  - $b \in N(N_2)$; choose $a \in N_2$ adjacent to b

Degree-safe components



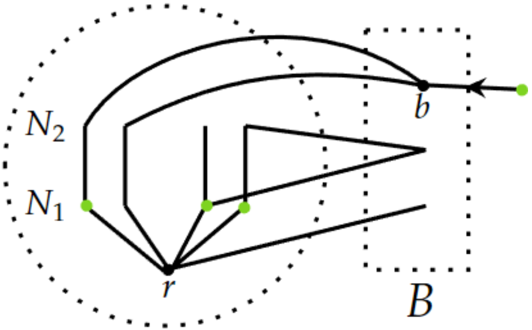
- **Case** $b \notin N(N_1)$:
  - $b \in N(N_2)$; choose $a \in N_2$ adjacent to b
  - slide a token from branch a to any other empty one, if needed

Degree-safe components



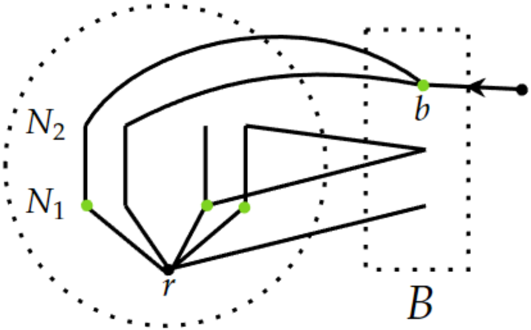
- **Case** $b \notin N(N_1)$:
  - $b \in N(N_2)$; choose $a \in N_2$ adjacent to b
  - slide a token from branch a to any other empty one, if needed

Degree-safe components



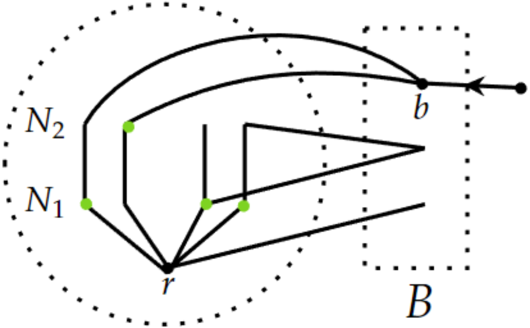
- **Case** $b \notin N(N_1)$:
  - $b \in N(N_2)$; choose $a \in N_2$ adjacent to b
  - slide a token from branch a to any other empty one, if needed
  - slide all tokens to N_1 of their respective branches

Degree-safe components



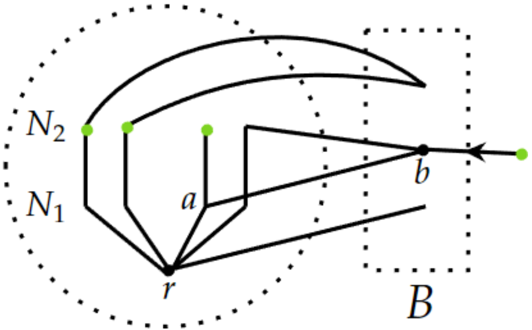
- **Case** $b \notin N(N_1)$:
  - $b \in N(N_2)$; choose $a \in N_2$ adjacent to b
  - slide a token from branch a to any other empty one, if needed
  - slide all tokens to N_1 of their respective branches
  - slide $b \to a$

Degree-safe components



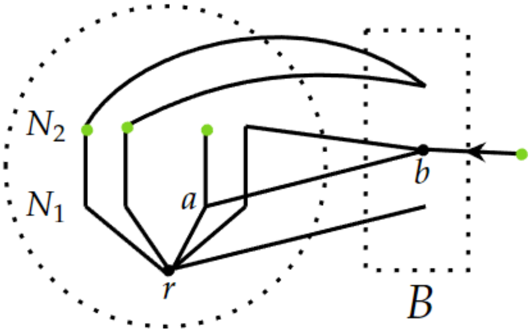
- **Case** $b \notin N(N_1)$:
  - $b \in N(N_2)$; choose $a \in N_2$ adjacent to b
  - slide a token from branch a to any other empty one, if needed
  - slide all tokens to N_1 of their respective branches
  - slide $b \to a$

Degree-safe components



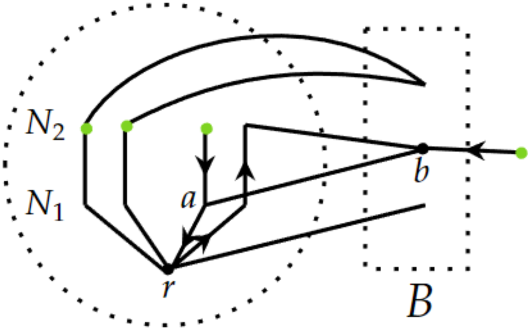
- **Case** $b \in N(N_1)$:

Degree-safe components



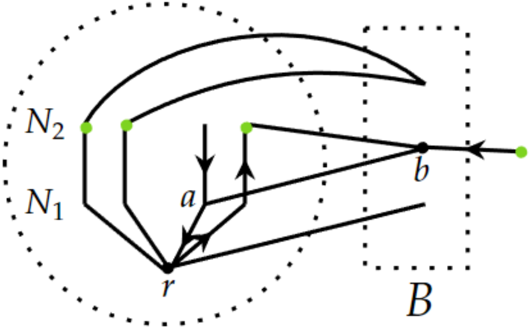
- **Case** $b \in N(N_1)$:
  - unique $a \in N_1$ is adjacent to b, as G has no 4-cycles

Degree-safe components



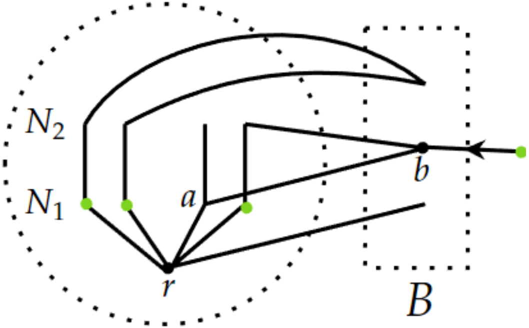
- **Case** $b \in N(N_1)$:
  - unique $a \in N_1$ is adjacent to b, as G has no 4-cycles
  - slide a token from branch a to any other empty one, if needed

Degree-safe components



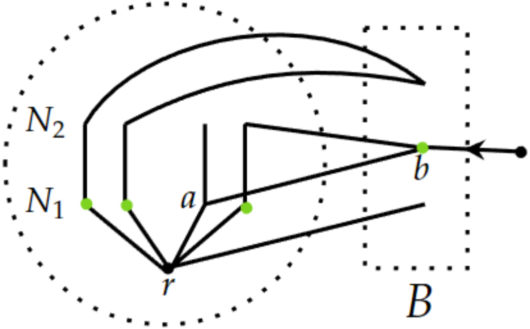
- **Case** $b \in N(N_1)$:
  - unique $a \in N_1$ is adjacent to b, as G has no 4-cycles
  - slide a token from branch a to any other empty one, if needed

Degree-safe components



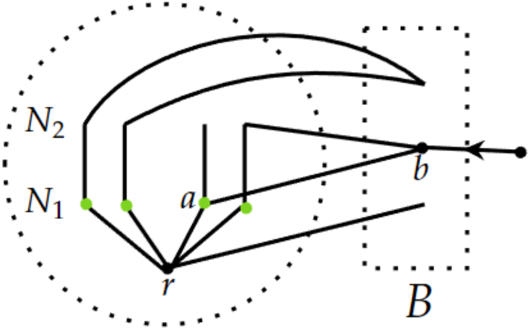
- **Case** $b \in N(N_1)$:
  - unique $a \in N_1$ is adjacent to b, as G has no 4-cycles
  - slide a token from branch a to any other empty one, if needed
  - slide all tokens to N_1 of their respective branches

Degree-safe components



- **Case** $b \in N(N_1)$:
  - unique $a \in N_1$ is adjacent to b, as G has no 4-cycles
  - slide a token from branch a to any other empty one, if needed
  - slide all tokens to N_1 of their respective branches
  - slide $b \to a$

Degree-safe components



- **Case** $b \in N(N_1)$:
  - unique $a \in N_1$ is adjacent to b, as G has no 4-cycles
  - slide a token from branch a to any other empty one, if needed
  - slide all tokens to N_1 of their respective branches
  - slide $b \to a$

Degree-safe components

- The step right before a token exits B - repeat the above procedure in reverse

Corollary

Let C be a degree-safe component. If a sequence from I_s to I_t exists, then also a sequence such that $N(C)$ never has > 1 tokens exists.

Proof.

Follow the path P from $c \in N(C)$ to r and apply the previous lemma when c enters B. We can always find such a path P that $N[P]$ contains no tokens, because all of them have been absorbed by C. □

Degree-safe components

- The step right before a token exits B - repeat the above procedure in reverse
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely inside $A \cup B$ - ignore it

Corollary

Let C be a degree-safe component. If a sequence from I_s to I_t exists, then also a sequence such that $N(C)$ never has > 1 tokens exists.

Proof.

Follow the path P from $c \in N(C)$ to r and apply the previous lemma when c enters B. We can always find such a path P that $N[P]$ contains no tokens, because all of them have been absorbed by C. □

Degree-safe components

- The step right before a token exits B - repeat the above procedure in reverse
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely inside $A \cup B$ - ignore it
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely outside $A \cup B$ - copy it

Corollary

Let C be a degree-safe component. If a sequence from I_s to I_t exists, then also a sequence such that $N(C)$ never has > 1 tokens exists.

Proof.

Follow the path P from $c \in N(C)$ to r and apply the previous lemma when c enters B. We can always find such a path P that $N[P]$ contains no tokens, because all of them have been absorbed by C. □

Degree-safe components

- The step right before a token exits B - repeat the above procedure in reverse
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely inside $A \cup B$ - ignore it
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely outside $A \cup B$ - copy it
- This gives us a sequence $\hat{I}'$ from I_s to I_t in which B never has more than 1 token

Corollary

Let C be a degree-safe component. If a sequence from I_s to I_t exists, then also a sequence such that $N(C)$ never has > 1 tokens exists.

Proof.

Follow the path P from $c \in N(C)$ to r and apply the previous lemma when c enters B. We can always find such a path P that $N[P]$ contains no tokens, because all of them have been absorbed by C. □

Diameter-safe components

Definition

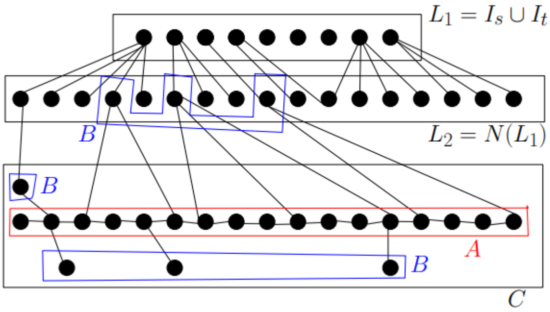
C is diameter-safe if $diam(G[C]) > k^3$

Diameter path A of a diameter-safe component C is the longest shortest path $u \to v$ in C.

Diameter-safe components

Lemma

Let A be a diameter path of C, and $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

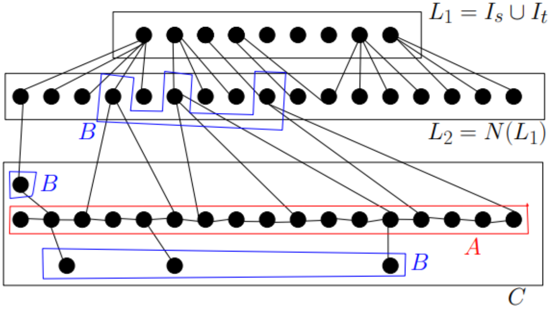


Diameter-safe components

Lemma

Let A be a diameter path of C, and $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$

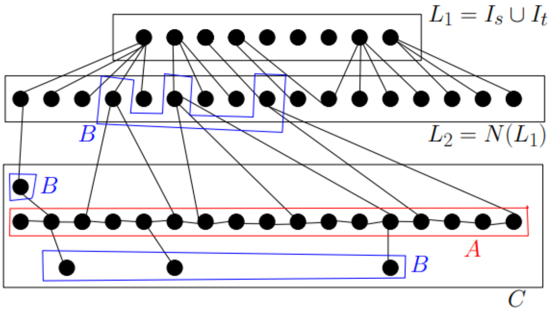


Diameter-safe components

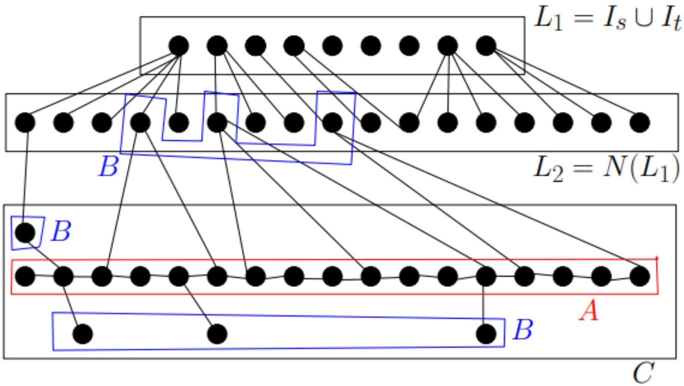
Lemma

Let A be a diameter path of C, and $B = N_G(A)$. If a sequence from I_s to I_t exists, then also a sequence such that B never has > 1 tokens exists.

- Note that $I_s \cap B = I_t \cap B = \emptyset$
- Like before, we will convert a sequence $\hat{I}$ from I_s to I_t into a sequence $\hat{I}'$ such that
  1. B never has > 1 tokens in $\hat{I}'_i$,
  2. at any step #tokens in $A \cup B$ is same and
  3. positions of tokens in $\overline{A \cup B}$ are same

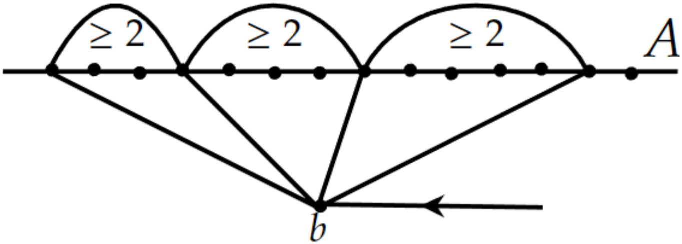


Diameter-safe components



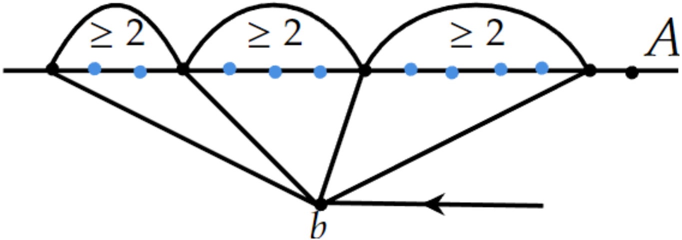
- No 2 non-consecutive vertices in A are adjacent, as A is a shortest path
- Consider step I_{i-1}, right before some token enters vertex $b \in B$

Diameter-safe components



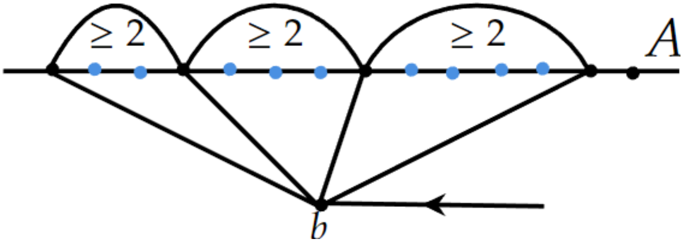
- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles

Diameter-safe components



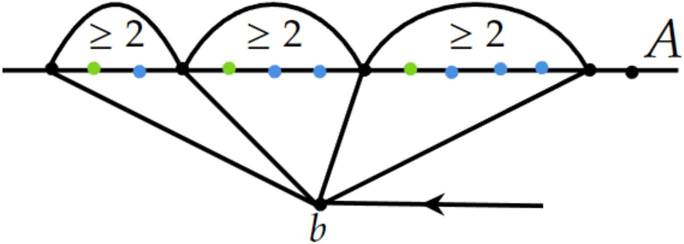
- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals

Diameter-safe components



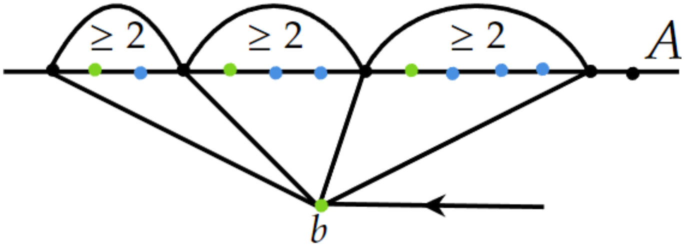
- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| > k$:

Diameter-safe components



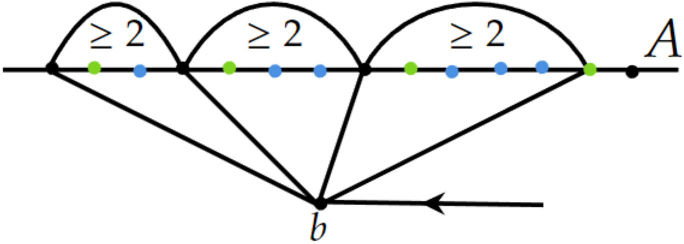
- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| > k$:
  - slide all tokens in A to first $\leq k - 1$ gap intervals 1-to-1

Diameter-safe components



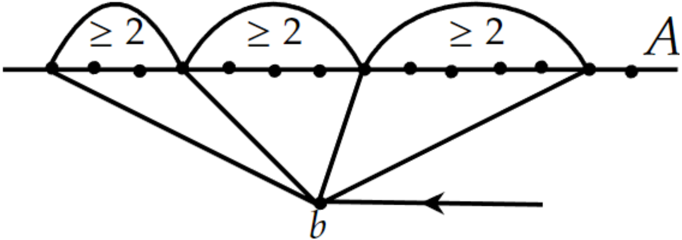
- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| > k$:
  - slide all tokens in A to first $\leq k - 1$ gap intervals 1-to-1
  - slide $b \rightarrow$ next neighbour of b in A

Diameter-safe components



- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| > k$:
  - slide all tokens in A to first $\leq k - 1$ gap intervals 1-to-1
  - slide $b \rightarrow$ next neighbour of b in A

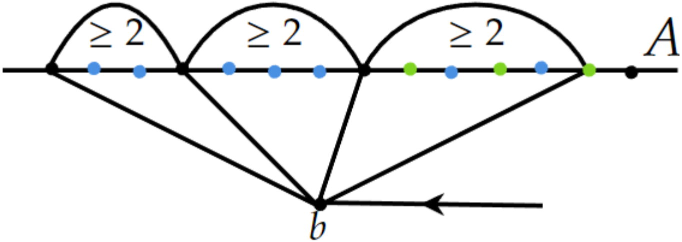
Diameter-safe components



- $\forall x, y \in N_A(b)$ $d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| \leq k$: then there are $\leq k-1$ gap intervals $+$ maybe 2 intervals at both ends of A. Hence there exists an interval i of size at least

$$\alpha = \frac{\mathrm{diam}(C) - |N_A(b)|}{|N_A(b)| + 1} \geq \frac{k^3 - k}{k+1} \geq 2k$$

Diameter-safe components

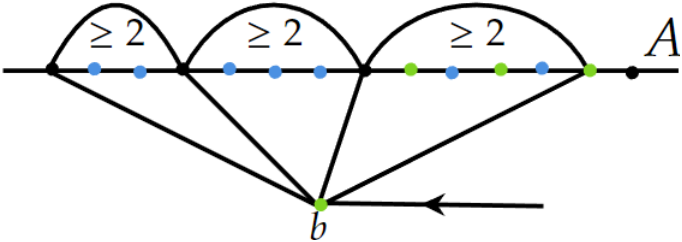


- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| \leq k$: then there are $\leq k - 1$ gap intervals + maybe 2 intervals at both ends of A. Hence there exists an interval i of size at least

$$\alpha = \frac{\mathrm{diam}(C) - |N_A(b)|}{|N_A(b)| + 1} \geq \frac{k^3 - k}{k + 1} \geq 2k$$

  - slide all tokens ($\leq k - 1$) in A to interval i

Diameter-safe components

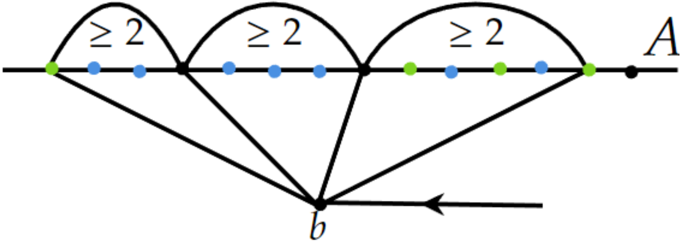


- $\forall x, y \in N_A(b)\ d_A(x, y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| \leq k$: then there are $\leq k-1$ gap intervals + maybe 2 intervals at both ends of A. Hence there exists an interval i of size at least

$$\alpha = \frac{\mathrm{diam}(C) - |N_A(b)|}{|N_A(b)| + 1} \geq \frac{k^3 - k}{k+1} \geq 2k$$

  - slide all tokens ($\leq k-1$) in A to interval i
  - slide $b \to$ any neighbour of b in A

Diameter-safe components



- $\forall x, y \in N_A(b)\ d_A(x,y) \geq 3$, as G has no 4-cycles
- We call the vertices between consecutive neighbours of b gap intervals
- **Case** $|N_A(b)| \leq k$: then there are $\leq k-1$ gap intervals + maybe 2 intervals at both ends of A. Hence there exists an interval i of size at least

$$\alpha = \frac{\text{diam}(C) - |N_A(b)|}{|N_A(b)| + 1} \geq \frac{k^3 - k}{k+1} \geq 2k$$

  - slide all tokens ($\leq k-1$) in A to interval i
  - slide $b \to$ any neighbour of b in A

Diameter-safe components

- The step right before a token exits B - repeat the above procedure in reverse

Diameter-safe components

- The step right before a token exits B - repeat the above procedure in reverse
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely inside $A \cup B$ - ignore it

Diameter-safe components

- The step right before a token exits B - repeat the above procedure in reverse
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely inside $A \cup B$ - ignore it
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely outside $A \cup B$ - copy it

Diameter-safe components

- The step right before a token exits B - repeat the above procedure in reverse
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely inside $A \cup B$ - ignore it
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely outside $A \cup B$ - copy it
- This gives us a sequence $\hat{I}'$ from I_s to I_t in which B never has more than 1 token

Diameter-safe components

- The step right before a token exits B - repeat the above procedure in reverse
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely inside $A \cup B$ - ignore it
- If a move $\hat{I}_i \to \hat{I}_{i+1}$ is completely outside $A \cup B$ - copy it
- This gives us a sequence $\hat{I}'$ from I_s to I_t in which B never has more than 1 token

Corollary

Let C be a diameter-safe component. If a sequence from I_s to I_t exists, then also a sequence such that $N(C)$ never has > 1 tokens exists.

Proof.

Follow the path P from $c \in N(C)$ to the closest vertex in diameter-path A and apply the previous lemma when c enters B. □

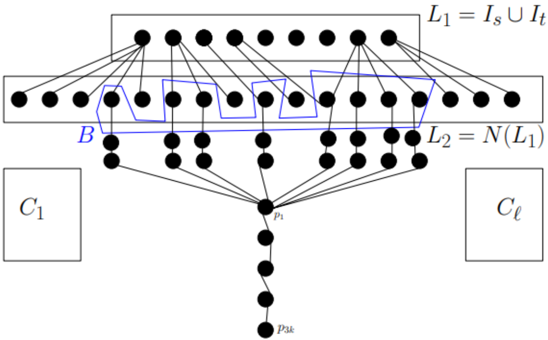
Safe components: replacement gadget

Lemma

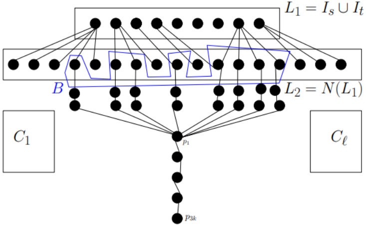
Let C be a safe component in $G[L_3]$ and G' be the graph obtained from G as follows:

- *delete C*
- *$\forall v \in N(C)$ add new vertices $v \to v' \to v''$*
- *add a path $p_1, \ldots p_{3k}$*
- *add edges $v'' \to p_1$*

Then $(G, I_s, I_t) \equiv (G', I_s, I_t)$.



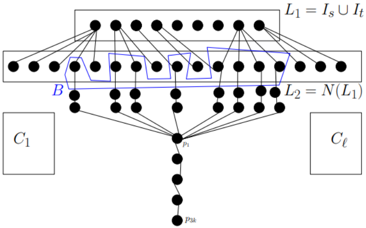
Safe components: replacement gadget



Proof.

- From previous corollaries, there exist a sequence $\hat{I}$ from I_s to I_t with at most 1 token in $N(C)$ at all times

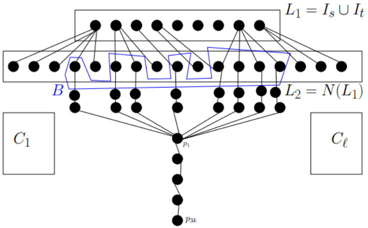
Safe components: replacement gadget



Proof.

- From previous corollaries, there exist a sequence $\hat{I}$ from I_s to I_t with at most 1 token in $N(C)$ at all times
- The replacement component can absorb/project k tokens

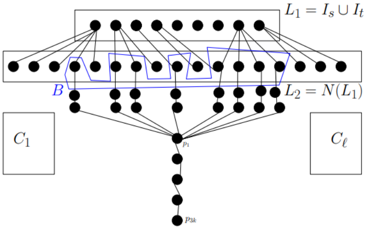
Safe components: replacement gadget



Proof.

- From previous corollaries, there exist a sequence $\hat{I}$ from I_s to I_t with at most 1 token in $N(C)$ at all times
- The replacement component can absorb/project k tokens
- So we can mimic $\hat{I}$ in the replacement component

Safe components: replacement gadget



Proof.

- From previous corollaries, there exist a sequence $\hat{I}$ from I_s to I_t with at most 1 token in $N(C)$ at all times
- The replacement component can absorb/project k tokens
- So we can mimic $\hat{I}$ in the replacement component
- Note that size of replacement component is $3k + 2|N(C)|$, and $N(C) \subseteq L_2$

Bounding the size of G

Lemma

Assume $u \in L_1$ *with* $\deg(u) > 2k^2$ *(WLOG* $u \in I_s$*). Then there exists* I'_s *such that* $I_s \Delta I'_s = \{u, u'\} \in E(G)$ *and* $\deg(u') \leq 2k^2$*.*

Bounding the size of G

Lemma

Assume $u \in L_1$ *with* $\deg(u) > 2k^2$ *(WLOG* $u \in I_s$*). Then there exists* I'_s *such that* $I_s \Delta I'_s = \{u, u'\} \in E(G)$ *and* $\deg(u') \leq 2k^2$.

We can apply this lemma exhaustively until $\Delta(L_1) \leq 2k^2$.

Bounding the size of G

Lemma

Assume $u \in L_1$ with $\deg(u) > 2k^2$ (WLOG $u \in I_s$). Then there exists I'_s such that $I_s \Delta I'_s = \{u, u'\} \in E(G)$ and $\deg(u') \leq 2k^2$.

We can apply this lemma exhaustively until $\Delta(L_1) \leq 2k^2$.

Corollary

For (G, I_s, I_t), there exists equivalent (G', I_s, I_t) such that $|L_2| \leq 4k^3$, and each safe component is replaced in G' by replacement component of size at most $3k + 8k^3 = O(k^3)$

Bounding the size of G

Lemma

Assume $u \in L_1$ with $\deg(u) > 2k^2$ (WLOG $u \in I_s$). Then there exists I_s' such that $I_s \Delta I_s' = \{u, u'\} \in E(G)$ and $\deg(u') \leq 2k^2$.

We can apply this lemma exhaustively until $\Delta(L_1) \leq 2k^2$.

Corollary

For (G, I_s, I_t), there exists equivalent (G', I_s, I_t) such that $|L_2| \leq 4k^3$, and each safe component is replaced in G' by replacement component of size at most $3k + 8k^3 = O(k^3)$

Lemma

In G', every bad component has at most $k^{O(k^3)}$ vertices.

Bounding the size of G

Lemma

Assume $u \in L_1$ with $\deg(u) > 2k^2$ (WLOG $u \in I_s$). Then there exists I'_s such that $I_s \Delta I'_s = \{u, u'\} \in E(G)$ and $\deg(u') \leq 2k^2$.

We can apply this lemma exhaustively until $\Delta(L_1) \leq 2k^2$.

Corollary

For (G, I_s, I_t), there exists equivalent (G', I_s, I_t) such that $|L_2| \leq 4k^3$, and each safe component is replaced in G' by replacement component of size at most $3k + 8k^3 = O(k^3)$

Lemma

In G', every bad component has at most $k^{O(k^3)}$ vertices.

From now on, we refer to both bounded and bad components as bounded components.

Bounding the size of G

It remains to bound the *number* of safe and bounded components.

Bounding the size of G

It remains to bound the *number* of safe and bounded components.

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Bounding the size of G

It remains to bound the *number* of safe and bounded components.

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Lemma

Let S_1 and S_2 be equivalent safe components. Then $(G, I_s, I_t) \equiv (G - S_2, I_s, I_t)$.

Bounding the size of G

It remains to bound the *number* of safe and bounded components.

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Lemma

Let S_1 and S_2 be equivalent safe components. Then $(G, I_s, I_t) \equiv (G - S_2, I_s, I_t)$.

Lemma

Let $B_1, \ldots, B_{k+1}$ be equivalent bounded components. Then $(G, I_s, I_t) \equiv (G - B_1, I_s, I_t)$.

Bounding the size of G

It remains to bound the *number* of safe and bounded components.

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Lemma

Let S_1 and S_2 be equivalent safe components. Then $(G, I_s, I_t) \equiv (G - S_2, I_s, I_t)$. Let $B_1, \ldots, B_{k+1}$ be equivalent bounded components. Then $(G, I_s, I_t) \equiv (G - B_1, I_s, I_t)$.

We can exhaustively remove equivalent components, and obtain

Bounding the size of G

It remains to bound the *number* of safe and bounded components.

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Lemma

Let S_1 and S_2 be equivalent safe components. Then $(G, I_s, I_t) \equiv (G - S_2, I_s, I_t)$. Let $B_1, \ldots, B_{k+1}$ be equivalent bounded components. Then $(G, I_s, I_t) \equiv (G - B_1, I_s, I_t)$.

We can exhaustively remove equivalent components, and obtain

Corollary

G has at most $\sigma(G)$ safe components and $k\beta(G)$ bounded components.

Bounding the size of G

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Corollary

G has at most $\sigma(G)$ safe components and $k\beta(G)$ bounded components.

Bounding the size of G

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Corollary

G has at most $\sigma(G)$ safe components and $k\beta(G)$ bounded components.

Lemma

- $\sigma(G) = 2^{O(k^6)}$ *and* $\beta(G) = 2^{k^{O(k^3)}}$

Bounding the size of G

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Corollary

G has at most $\sigma(G)$ safe components and $k\beta(G)$ bounded components.

Lemma

- $\sigma(G) = 2^{O(k^6)}$ *and* $\beta(G) = 2^{k^{O(k^3)}}$
- $|L_3| \leq k^3 \cdot 2^{O(k^6)} + k^{O(k^3)} \cdot 2^{k^{O(k^3)}}$

Bounding the size of G

Definition

Let C_1 and C_2 be components in $G[L_3]$, and $D_i = N(C_i)$. We call C_1 and C_2 equivalent if $D_1 = D_2 = D$ and $G[N[C_1]]$ is isomorphic to $G[N[C_2]]$ by an isomorphism that fixes D point-wise. Let $\beta(G)$ be the number of equivalence classes of bounded components, and $\sigma(G)$ - of safe components.

Corollary

G has at most $\sigma(G)$ safe components and $k\beta(G)$ bounded components.

Lemma

- $\sigma(G) = 2^{O(k^6)}$ *and* $\beta(G) = 2^{k^{O(k^3)}}$
- $|L_3| \leq k^3 \cdot 2^{O(k^6)} + k^{O(k^3)} \cdot 2^{k^{O(k^3)}}$
- $|V(G)| = |L_1| + |L_2| + |L_3| = 2^{k^{O(k^3)}}$

Summary

1. While $\Delta(L_1) > 2k^2$, equivalently modify I_s or I_t

Summary

1. While $\Delta(L_1) > 2k^2$, equivalently modify I_s or I_t
2. Replace all safe components with a small gadget

Summary

1. While $\Delta(L_1) > 2k^2$, equivalently modify I_s or I_t
2. Replace all safe components with a small gadget
3. Partition the components of L_3 into equivalence classes

Summary

1. While $\Delta(L_1) > 2k^2$, equivalently modify I_s or I_t
2. Replace all safe components with a small gadget
3. Partition the components of L_3 into equivalence classes
4. Remove redundant components from each class

Summary

1. While $\Delta(L_1) > 2k^2$, equivalently modify I_s or I_t
2. Replace all safe components with a small gadget
3. Partition the components of L_3 into equivalence classes
4. Remove redundant components from each class
5. Solve the resulting instance naively with e.g. BFS on its reconfiguration graph

Summary

1. While $\Delta(L_1) > 2k^2$, equivalently modify I_s or I_t
2. Replace all safe components with a small gadget
3. Partition the components of L_3 into equivalence classes
4. Remove redundant components from each class
5. Solve the resulting instance naively with e.g. BFS on its reconfiguration graph

Time complexity: $O(kn^3) + 2^{k^{O(k^3)}} + O(k^2\binom{f(k)}{k}^2)$

Summary

1. While $\Delta(L_1) > 2k^2$, equivalently modify I_s or I_t
2. Replace all safe components with a small gadget
3. Partition the components of L_3 into equivalence classes
4. Remove redundant components from each class
5. Solve the resulting instance naively with e.g. BFS on its reconfiguration graph

Time complexity: $O(kn^3) + 2^{k^{O(k^3)}} + O(k^2 \binom{f(k)}{k}^2)$

Corollary

TOKEN SLIDING is fixed-parameter tractable when parameterized by k on graphs of girth ≥ 5.

Open questions

It remains open

- whether Token Sliding admits a (polynomial) kernel on graphs of girth five or more

Open questions

It remains open

- whether Token Sliding admits a (polynomial) kernel on graphs of girth five or more
- whether the problem remains tractable if we forbid cycles of length $p mod q$

Open questions

It remains open

- whether Token Sliding admits a (polynomial) kernel on graphs of girth five or more
- whether the problem remains tractable if we forbid cycles of length $p mod q$
- whether the problem remains tractable if we exclude odd cycles

References

Valentin Bartier, Nicolas Bousquet, Jihad Hanna, Amer E. Mouawad and Sebastian Siebertz (2022)
Token sliding on graphs of girth five
arXiv

The End