Revisiting Token Sliding on Chordal Graphs

WG 2026 @ Kortrijk, Belgium

June 2, 2026

Prafullkumar Tale

Indian Institute of Science Education and Research, Pune, India

WG 2026

IISER PUNE

Joint work with

Rajat Adak

Saraswati Nanoti

Indian Institute of Sciences, Bangalore, India

(Independent Set) Token Sliding-Reachability

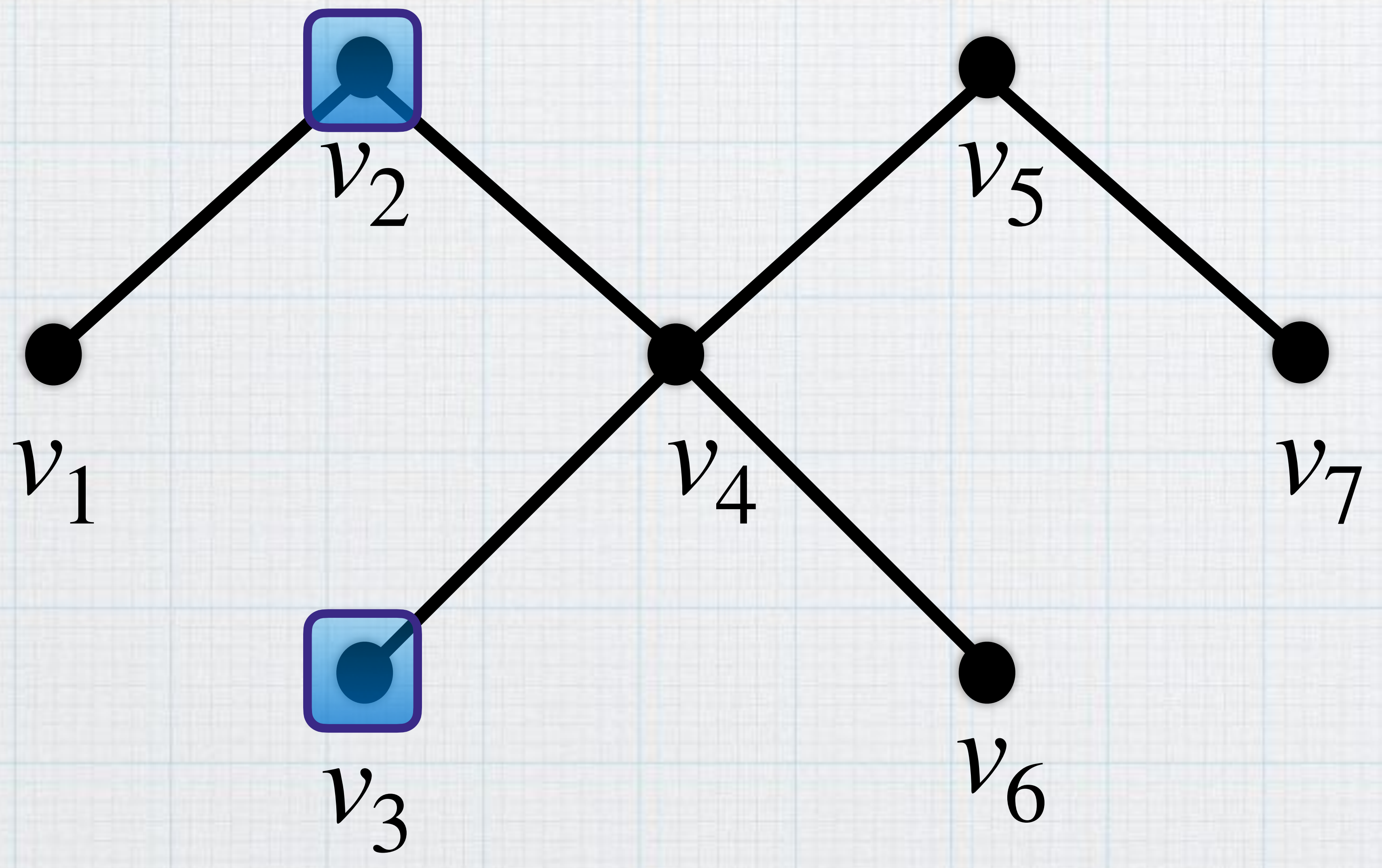
Input: Graph G, two independent sets S and T.

Output: Can we reach from configurations of token as in S to the configurations as in T by moving one token at a time such as every intermediate configuration is also an independent set?

(Independent Set) Token Sliding–Reachability

Input: Graph G, two independent sets S and T.

Output: Can we reach from configurations of token as in S to the configurations as in T by moving one token at a time such as every intermediate configuration is also an independent set?

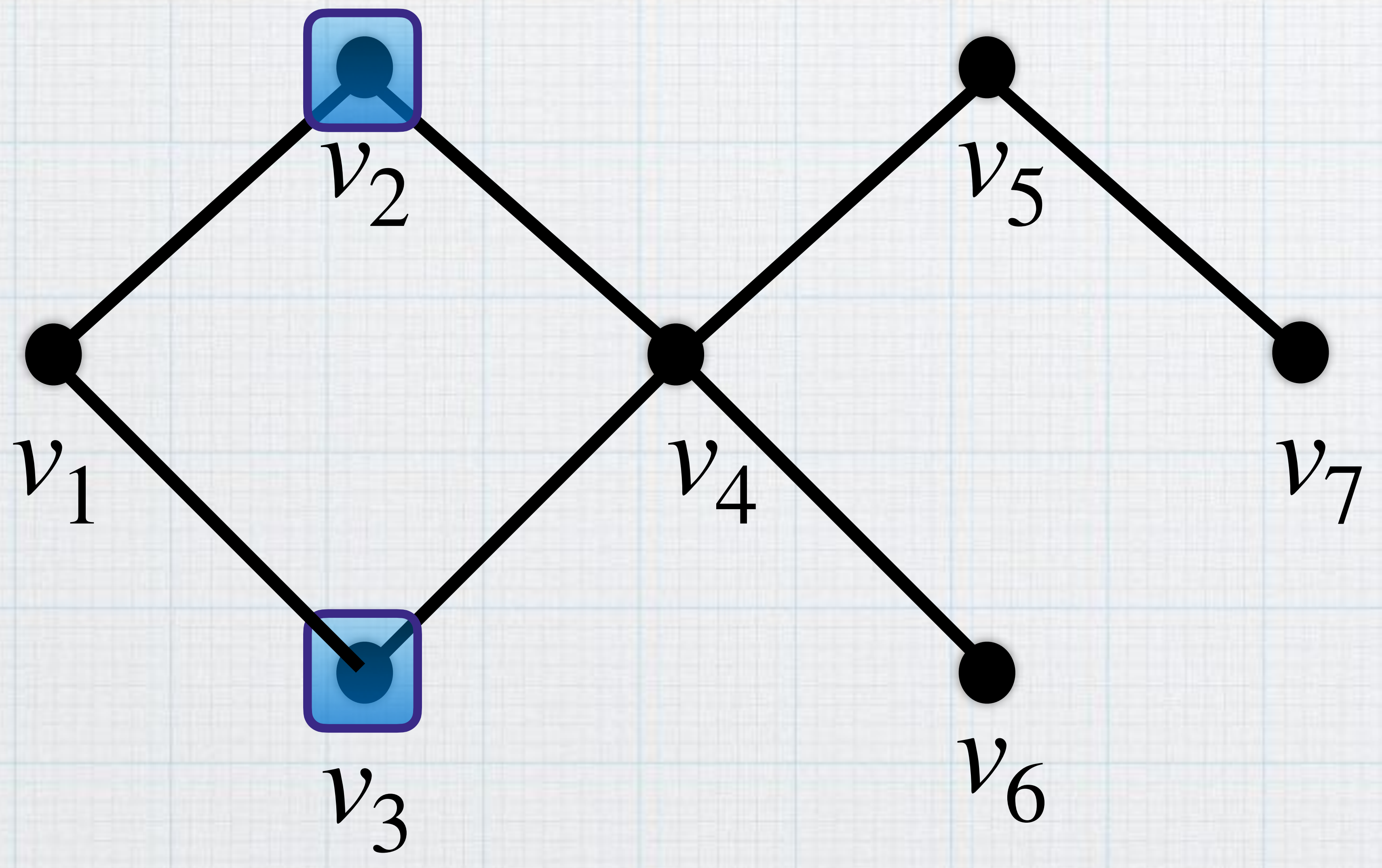


- $S = \{v_2, v_3\}, T = \{v_5, v_6\}$
- We can move only one token in a given round.

(Independent Set) Token Sliding–Connectivity

Input: Graph G, ~~two independent sets S and T~~.

Output: Can we reach from any given configuration to any other configuration moving one token at a time such as every intermediate configuration is also an independent set?



- No other independent set is readable from $\{v_2, v_3\}$.
- We can move only one token in a given round.

(Independent Set) Token Sliding-Connectivity

Input: Graph G, ~~two independent sets S and T~~.

Output: Can we reach from any given configuration to any other configuration moving one token at a time such as every intermediate configuration is also an independent set?

(Independent Set) Token Sliding-Connectivity

Input: Graph G and integer k.

Output: Is the k-TS-reconf. graph $TS(G)$ of G connected?

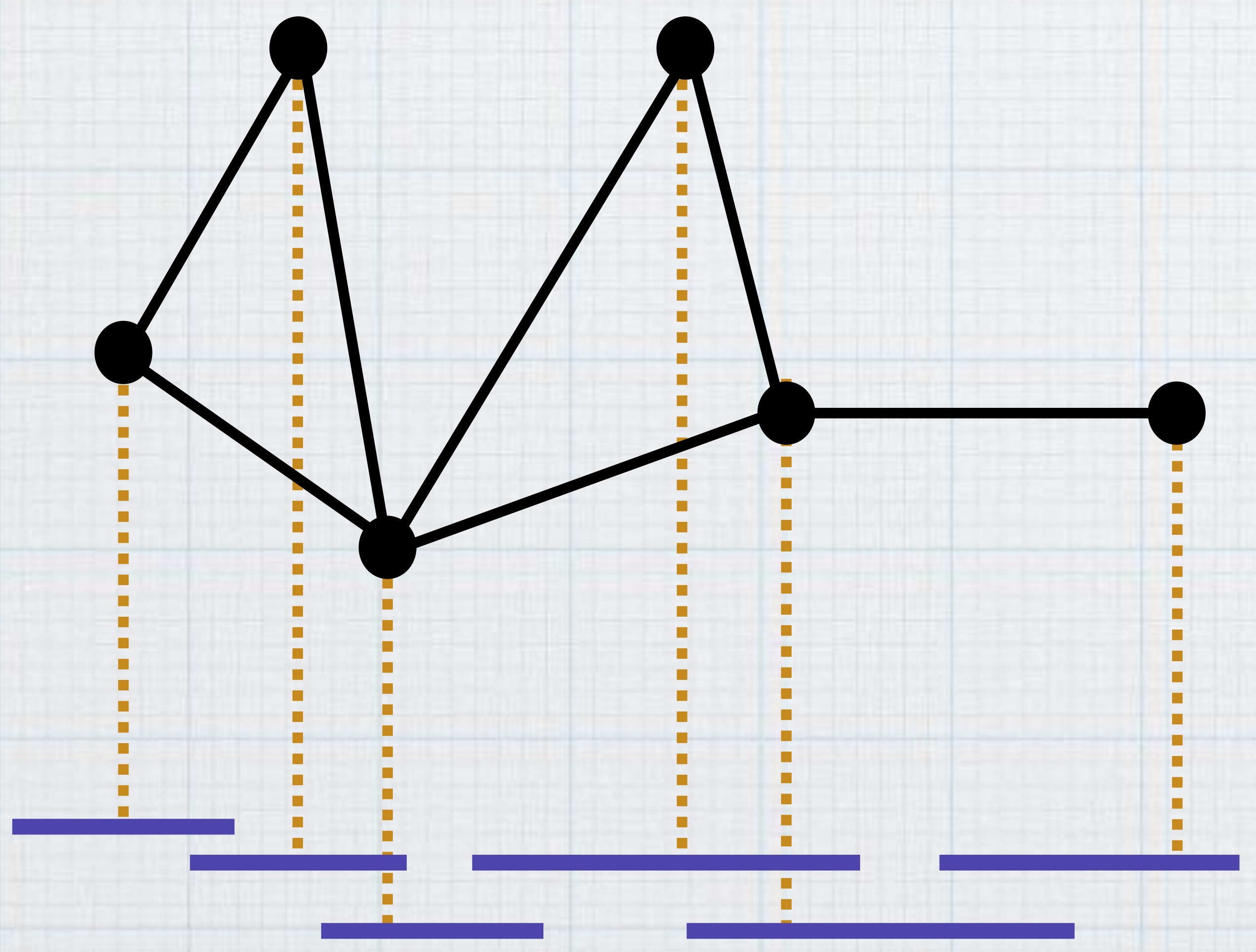
The vertices of $TS(G)$ are k-independent sets of G, and (I, J) is an edge of $TS(G)$ if and only if $J \setminus I = \{u\}$, $I \setminus J = v$ and (u, v) is an edge of G.

Bonamy and Bousquet in `Token Sliding on Chordal Graphs' [WG'17] proved the following.

Theorem. Token Sliding-Connectivity
- admits polynomial time algorithm on interval graphs, but
- is co-NP-hard for split graphs.

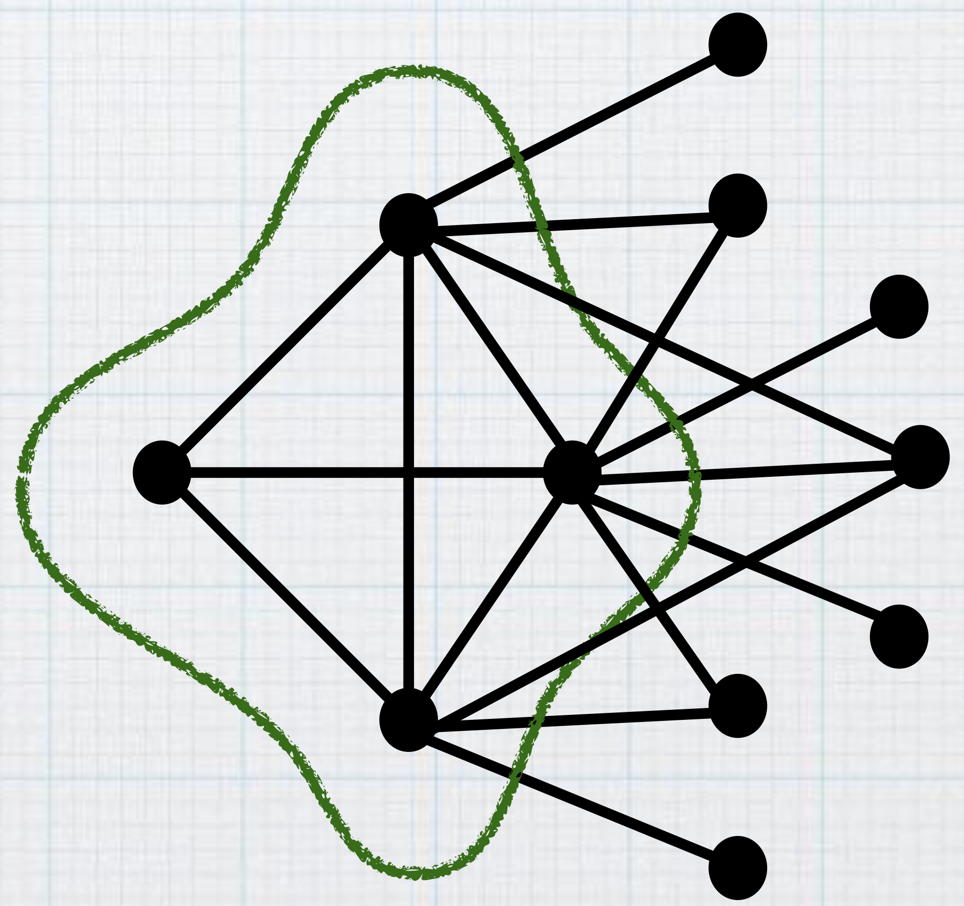
Interval Graphs

Vertex u corr. to an interval I_u on real-line. Edge uv iff $I_u \cap I_v \neq \emptyset$.



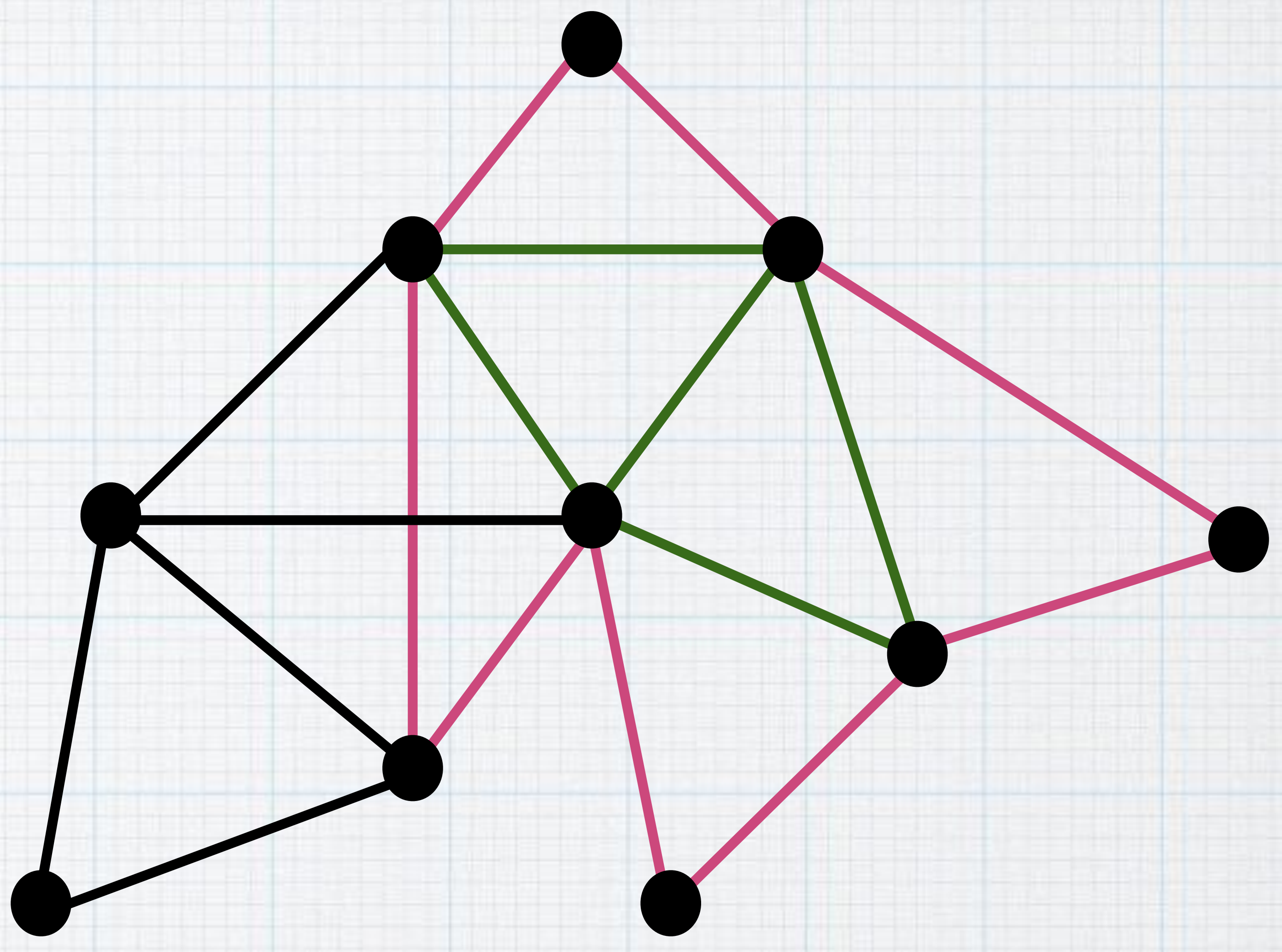
Split Graphs

Partition vertex set into a clique and an independent set.

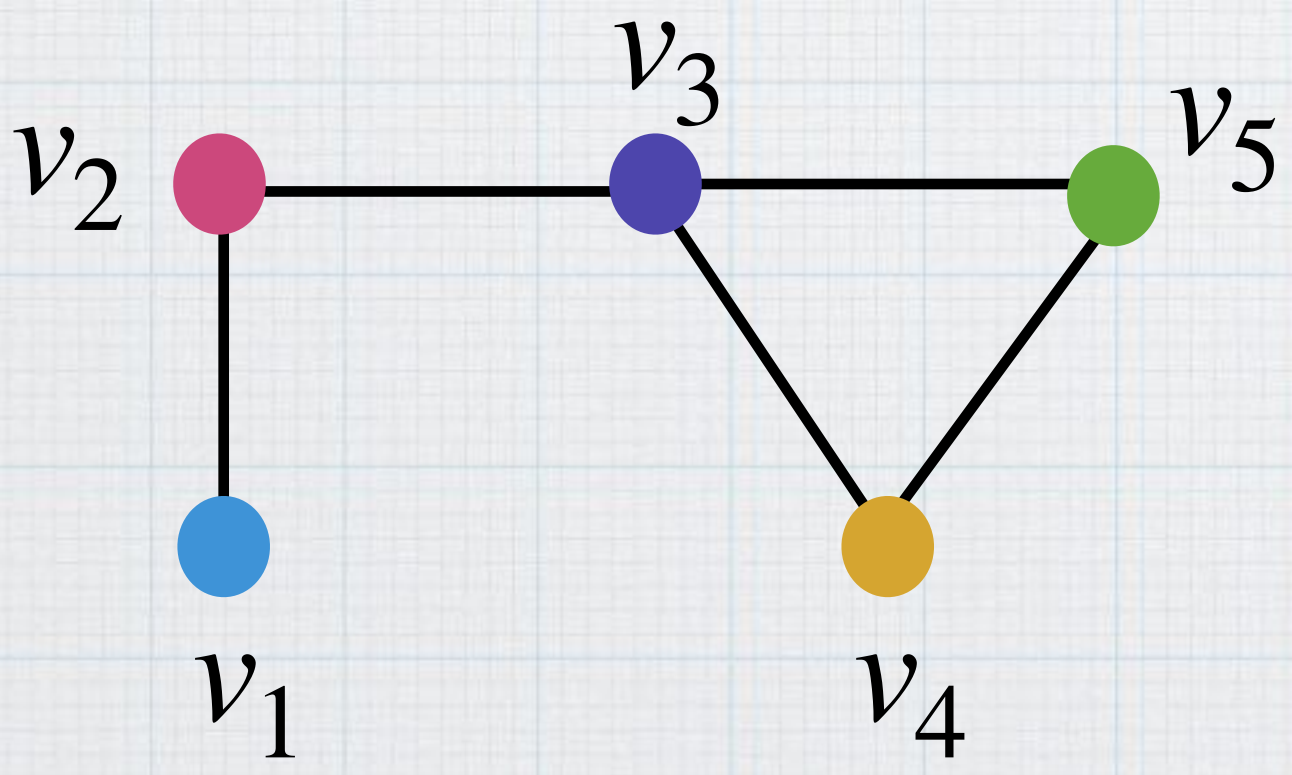


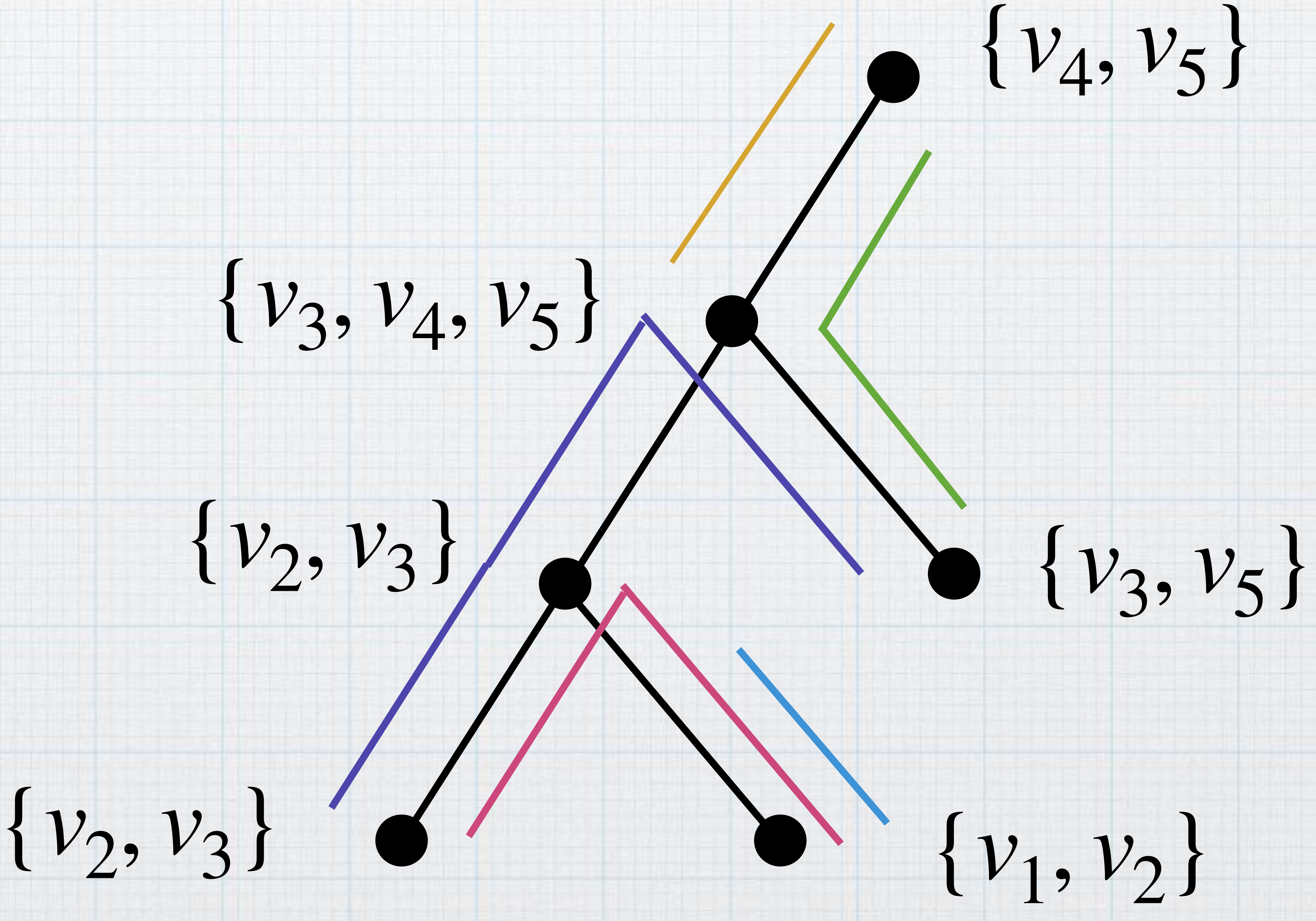
	Interval	Split
Dominating Set	poly	NP-hard
Conn Dom Set	poly	NP-hard
Steiner Tree	poly	NP-hard
Multicut with Deletable Terminals	poly	NP-hard
Subset Feedback Vertex Set	poly	NP-hard
Fire Break	poly	NP-hard
Graph Isomorphism	poly	Hard

Chordal Graphs: Every induced cycle of length at least four has a chord.



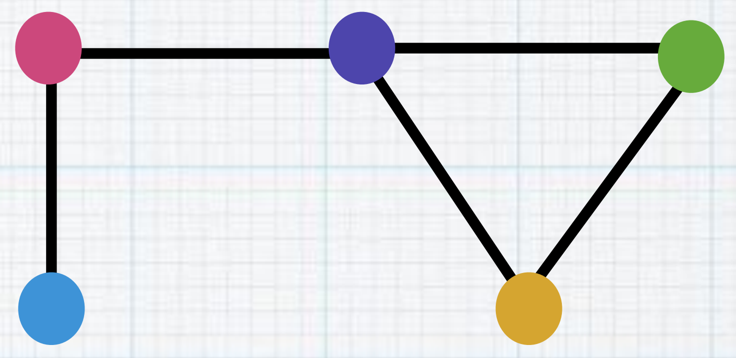
Clique-tree of a chordal graph: A tree whose vertices are the cliques of the graph, and for every vertex in the graph, the set of cliques that contain it form a subtree of clique-tree.

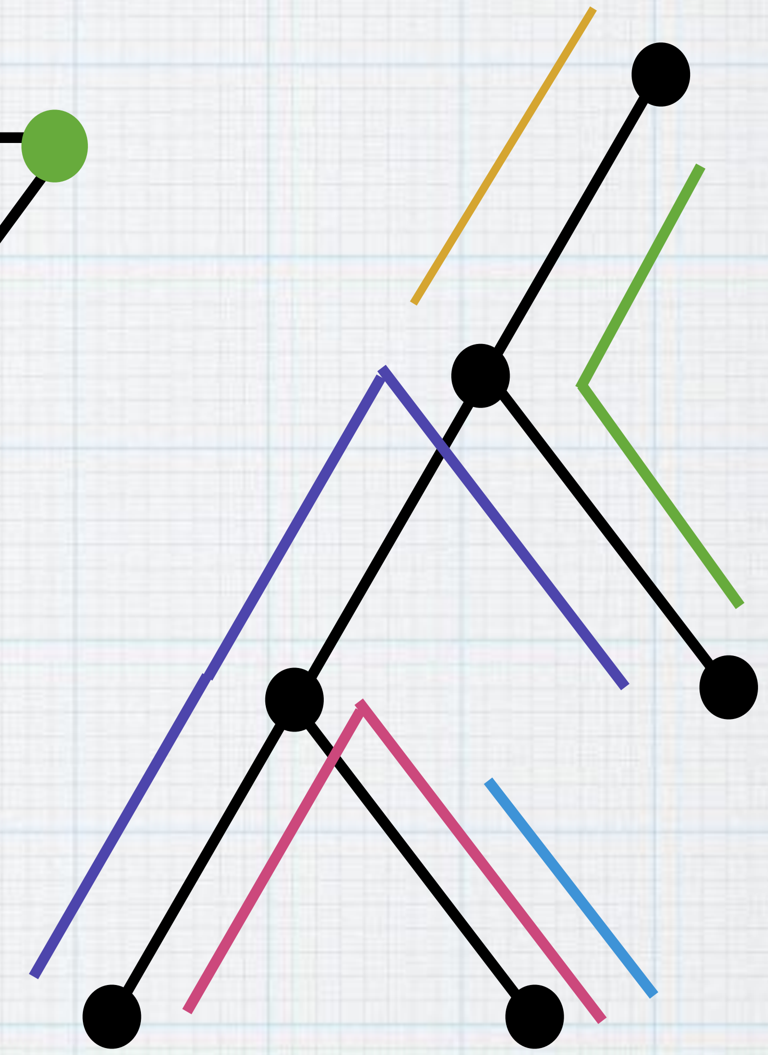




Chordal Graphs Trees

Interval Graphs Paths Split Graphs Stars





Clique tree for
- for an interval graph has maximum degree two
- for a split graph can have unbounded maximum degree.

Let d be a fixed constant. For an integer k and chordal graph G of the maximum clique-tree degree at most d, can the connectivity of $TS(G)$ be decided in polynomial time?
— Bonamy and Bousquet [WG'17]

Let d be a fixed constant. For an integer k and chordal graph G of the maximum clique-tree degree at most d, can the connectivity of $TS(G)$ be decided in polynomial time?
— Bonamy and Bousquet [WG'17]

Theorem 1. On chordal graphs of maximum clique-tree degree 4, Token Sliding Connectivity problem is co-NP-hard.

- Maximum clique-tree degree is not an ideal parameter.
- How about a larger parameter leafage?

$$l(G) = \min\{\#\ leaves(T) \mid T \text{ is a clique-tree of } G\}$$

- Habib and Stacho [ESA 2009] showed that we can compute the leafage of a connected chordal graph in polynomial time.

	Interval	Split	Para by leafage	
Dominating Set	poly	NP–hard	$2^{\mathcal{O}(l)} \cdot n^{\mathcal{O}(1)}$	[GMSST, Algo’24]
Conn Dom Set	poly	NP–hard		
Steiner Tree	poly	NP–hard		
Multicut with Deletable Terminals	poly	NP–hard	$n^{\mathcal{O}(l)}$	[GMSST, Algo’24]
Subset Feedback Vertex Set	poly	NP–hard	$n^{\mathcal{O}(l)}$	[PT, Algo’24]
Fire Break	poly	NP–hard	$n^{\mathcal{O}(l)}$	[BBEHPR, Networks’73]
Graph Isomorphism	poly	Hard	$2^{\mathcal{O}(l)} \cdot n^{\mathcal{O}(1)}$	[ANPZ, Algo’26]

Does Token Sliding Connectivity admit a fixed parameter tractable algorithm when parameterized by the leafage of the input chordal graph, i.e., algorithm running in time $f(l) \cdot n^{\mathcal{O}(1)}$?

Theorem 2. On chordal graphs, the Token Sliding Connectivity problem is co-W[1]-hard when parameterized by the leafage l of the input graph.

Does Token Sliding Connectivity admit an XP algorithm parameterized by the leafage of the input chordal graph, i.e. algorithm running in time $n^{f(l)}$?

- We belief it does but could not prove it.

- How about Token Sliding Reachability?

Theorem. Token Sliding Reachablity
– admits poly-time algorithm on interval graphs [BB, WG'17], but
– is PSPACE-complete for split graphs [BKLMOS, TCS'21].

Theorem 3. On chordal graphs of maximum clique-tree degree 3, Token Sliding Reachability problem is co-NP-hard.

Theorem 4. On chordal graphs, the Token Sliding Reachability problem is co-W[1]-hard when parameterized by the leafage l of the input graph.

Does Token Sliding Reachability admit an XP algorithm parameterized by the leafage of the input chordal graph?

- About the possible $n^{f(l)}$ algorithm for Token Sliding Reachability (and Token Sliding Connectivity)

- A non-trivial generalization of the algorithms
  – for trees by Demaine et al [ISAAC'14],
  – for interval graphs by Bonamy and Bousquet [WG'17].

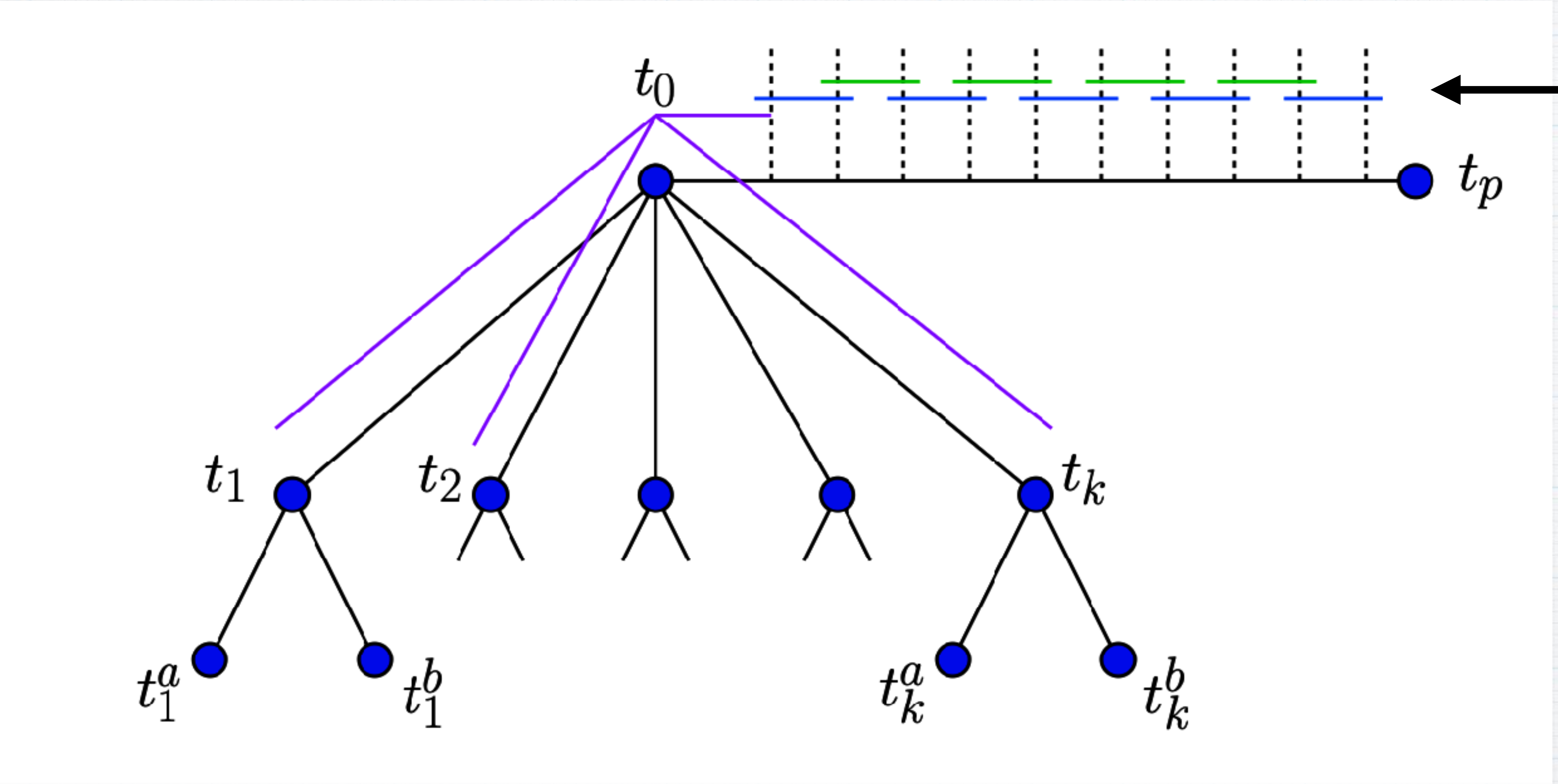
- Francis and Prabhakaran [FSTTCS'25] Token Sliding Reachability admits a poly- time algorithm on block graphs.

  Block graph: A graph in which every maximal 2-connected component is a clique.

Brief overview of the reduction

Multicolored Independent Set
Input: Graph H with partition $\langle V_1, V_2, \ldots, V_p \rangle$ of $V(H)$, where each part contains n vertices.
Output: Does there exist an independent set with exactly one vertex from each part?

- We construct an instance $(G, k = p \cdot n)$ of Token Sliding Connectivity such that G is a chordal graph and leafage of G is $2p + 1$.
- No-instance of MultiCol Ind-Set $\Rightarrow$ Yes-instance of TS-Conn.
- Yes-instance of MultiCol Ind-Set $\Rightarrow$ No-instance of TS-Conn.



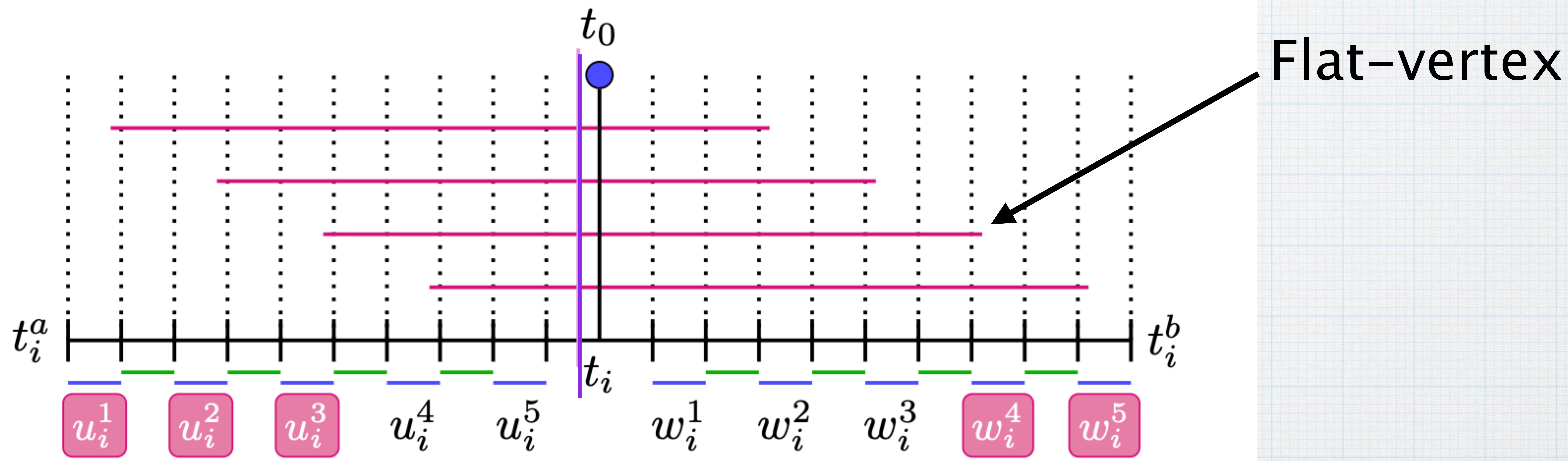
Parking structure consisting of $k \cdot n$ many blue vertices

Clique-tree of G

If H does **not** contain a k-multicol ind-set then any $k \cdot n$ sized ind-set in G, can reach to the parking structure.

If H contains a k-multicol ind-set X, then the corresponding $k \cdot n$ sized ind-set in G can **not** reach to the parking structure.

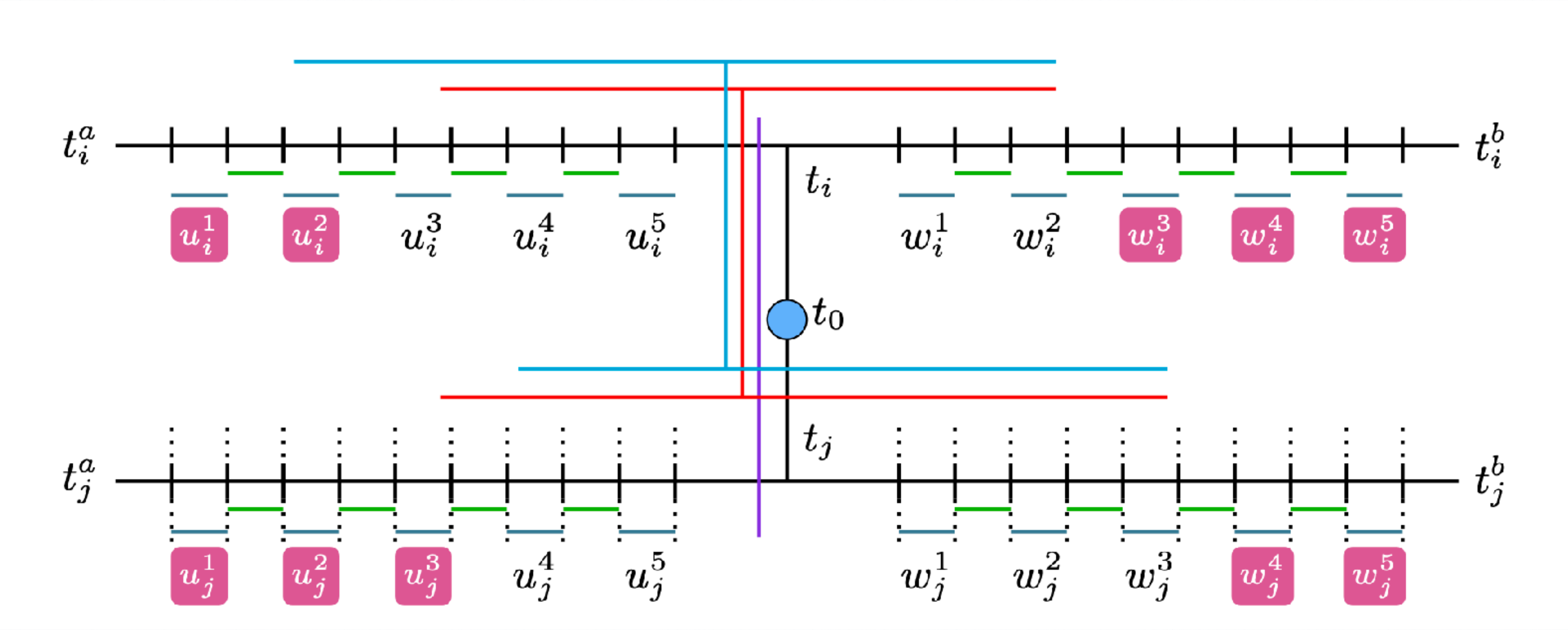
Each brach of there tree looks as follows:



Tokens are arranged as above iff $v_i^3 \in V_i$ is in X.

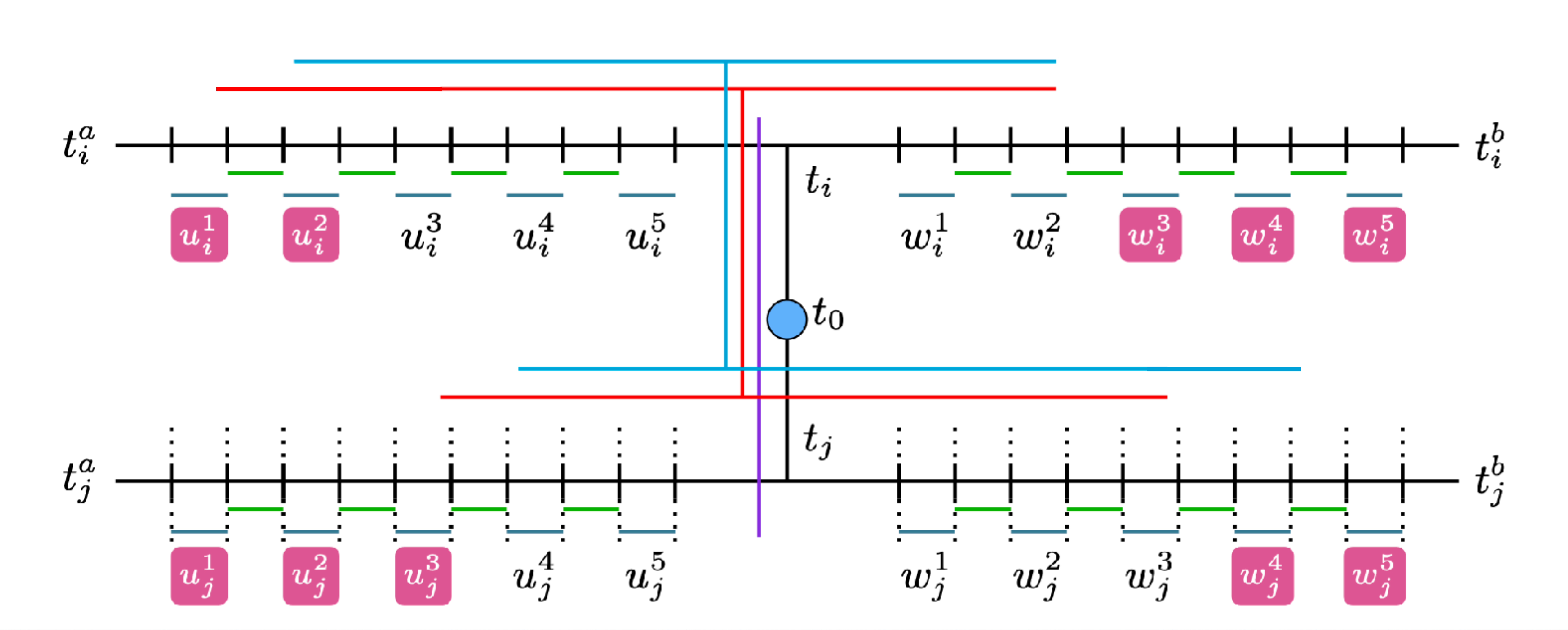
If the branch contains n tokens, then no flat-vertices can be used to move any token.

For $v_i^2 \in V_i$ and $v_j^3 \in V_j$ edge (v_i^2, v_j^3) is added as follows:

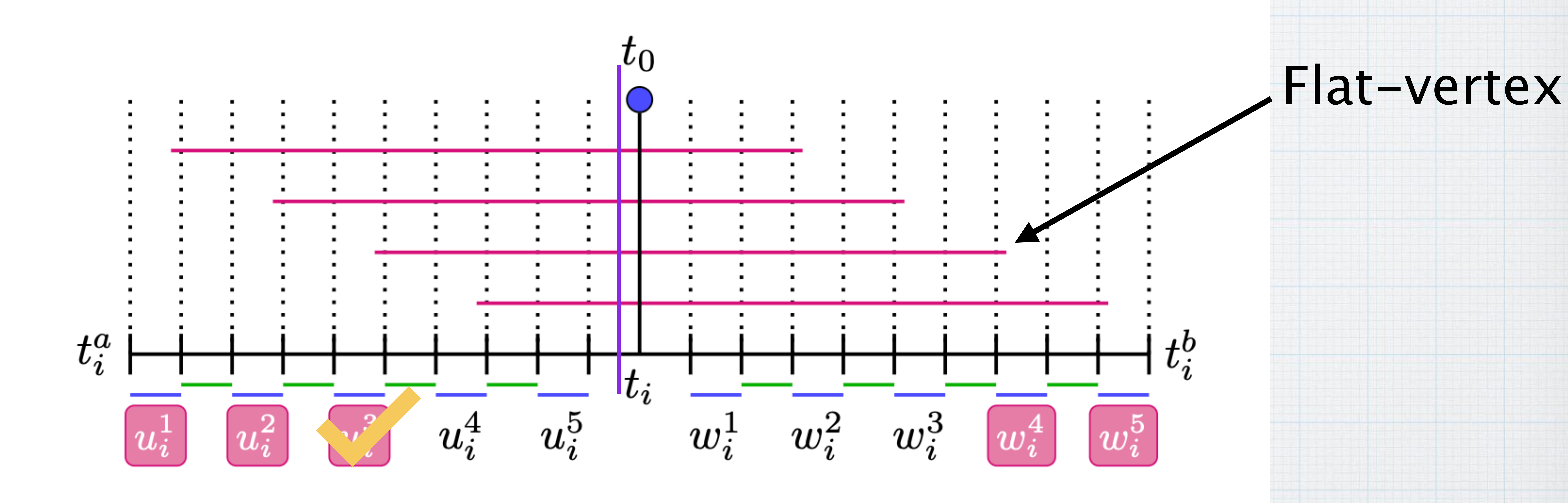


The edge structure allows token on u_i^2 to use cyan-vertex & token on u_j^3 can use red-vertex to reach t_0, and to the parking structure.

~~For $v_i^2 \in V_i$ and $v_j^3 \in V_j$ edge (v_i^2, v_j^3) is added as follows:~~



Edge with endpoint v_i^2 is forbidden by its another endpoint (like w_j^4), and hence can't use be used to move u_i^2 to t_0, and then to the parking structure.



Suppoe one of the vertex can be moved to the parking structure.

Now use flat-vertices to move other tokens to t_0 and then to the parking structure.

Any $k \cdot n$ sized independent set (of tokens) in G corresponds to set a X in graph H where X contains one vertex from each part in $\langle V_1, V_2, \ldots, V_p \rangle$.

If X is **not** an independent set, then edge structure between $v_i^2 \in V_i \cap X$ and $v_j^3 \in V_j \cap X$ allows one token in each brach to be moved to t_0 and then to the parking structure.

If X is an independent set, then the present configuration of tokens is in deadlock and can't be moved.

Since MultiCol Ind–Set is W[1]–hard para by k, we have TS–Conn is co–W[1]–hard para by leafage $l = 2k + 1$.

Open Questions

- Any positive results about Token Sliding Reachability or Token Sliding Connectivity.
  - $n^{f(l)}$ on chordal graphs?

- Following problems are poly-time solvable on internal graphs but NP-complete on split graph. Do they admit FPT or XP algorithm parameterized by leafage?

  Longest Path, Longest Cycle, Bandwidth, Cluster Vertex Deletion, Component Order Connectivity, etc.

Dank u! Merci! Danke!