Streamable Graphs

(Please note that this is a preliminary draft, and mistakes are possible.)

Contents

1 Preliminaries

▶ **Definition 1.** *Let* $G = (V, E)$ *with* $|V| = n$*. Let* $\sigma : V \mapsto [n]$ *be an ordering of the vertex of* G*. For each subset* $U \subset V$*, we use* $\sigma|_U$ *to denote the sub-ordering of* σ *induced by* U*. A set* $U \subset V$ *is streamed in* σ *if*

$$\left((\sigma|_U)^{-1}(1), (\sigma|_U)^{-1}(2), \dots, (\sigma|_U)^{-1}(|U|)\right) \tag{1}$$

is a path in G*.*

▶ **Definition 2.** *Let* $N(v)$ *denote the set of neighbors of* v *in* G*, and*

$$N^+_\sigma(v) := \{u \in N(v) : \sigma(u) > \sigma(v)\}. \tag{2}$$

We will refer to $N^+_\sigma(v)$ *as the succeeding neighbors of* v*.*

▶ **Definition 3.** *A graph* G *is streamable if there exists an ordering* $\sigma : V \mapsto [n]$ *such that for all* $v \in V$*,* $N^+_\sigma(v)$ *is streamed in* σ*. We call such ordering a streaming ordering of* G*.*

For example, all chordal graphs are streamable, as witnessed by any perfect elimination ordering.While C_4 is not streamable, W_4 is. Thus, being streamable is not hereditary.

2 The NP-completeness of recognition

▶ **Theorem 4.** *Recognition of streamable graphs is* NP*-complete.*

Clearly, the recognition of streamable graphs is in NP, as an ordering σ of the vertex set of the graph is a certificate. So we now only need to show the NP-hardness by reducing the following betweenness problem into the recognition of streamable graphs.

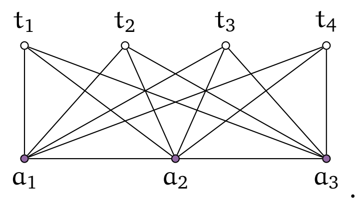
▶ **Definition 5 (Betweeness[1]).** *Given a finite set* S *and a set of ordered triples* $R \subset S \times S \times S$*, the betweenness problem on* (R, S) *asks to determine whether there exists a total ordering of S such that for every triple* (x, y, z) *in* R*, either* $x < y < z$ *or* $z < y < x$*.*

▶ **Theorem 6 ([1]).** *The betweenness problem is* NP*-complete.*

Now we introduce the following lemma that helps us to establish connection between the two problems:

▶ **Lemma 7.** *In a graph* $G = (V, E)$*, for any independent set* I *of size* k*, if there exists a set of vertices* $A \subset V$ *with* $N(v) = A$ *for all* $v \in I$ *and* $|A| \leqslant k - 1$*, then in any streaming ordering* σ *of* G*,* A *is streamed in* σ*.*

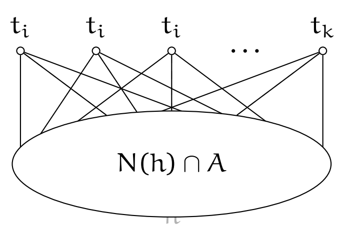
This lemma is somewhat ambiguous, and we refer to the following figure for an illustration:



For the graph above, Lemma 7 tells us $\{a_1, a_2, a_3\}$ must be streamd in σ. In this figure, this means either $\sigma(a_1) < \sigma(a_2) < \sigma(a_3)$ or $\sigma(a_3) < \sigma(a_2) < \sigma(a_1)$ since there are only two possible Hamiltonian paths in the subgraph induced by $\{a_1, a_2, a_3\}$. Now we proof Lemma 7.

Proof of Lemma 7. Given a streaming ordering σ, we let $h := \sigma^{-1}(1)$. We remark that $N(h) = N^+_\sigma(h)$, i.e., all neighbours of h are succeeding neighbours of it since it is the first one in the ordering, and thus $N(h)$ should be streamed in σ.

We now show $h \in I$ by contradiction. Assuming $h \in A$, h could have (open) neighbourhood of the following form:

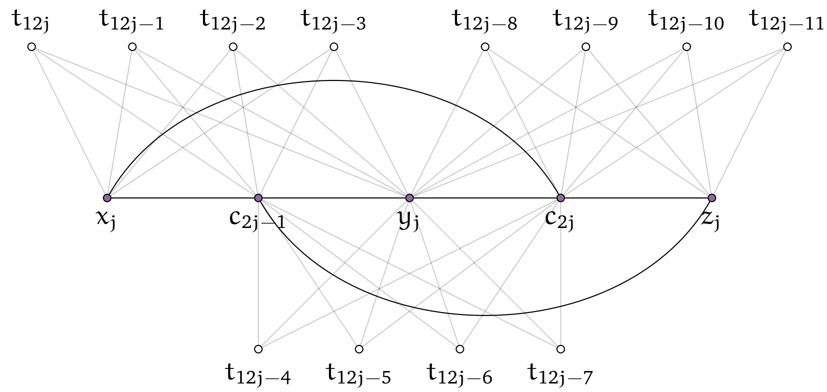


However, the neighbourhood is not possible to be streamed in any ordering as there is no Hamiltonian path, because in a Hamiltonian path, the $|N(h) \cap A|$ bottom vertices can only allow $|N(h) \cap A|+1 \leqslant |A| \leqslant k-1$ alternations between the two rows, while there are k vertices in the top row, making it impossible to for them to fit in any Hamiltonian path. Therefore, $h \in T$, and $N(h) = A$ should be streamed in σ. ◀

Now we describe the construction of graph $G_{R,S}$ given a betweenness problem (R, S) with $R := \{(x_j, y_j, z_j) : j \in [|R|], x_j, y_j, z_j \in S\}$.

1. Create the following sets of vertices:
   a. V_S of $|S|$ vertices, each of which corresponds to an element in S;
   b. V_C of $2|R|+|S|$ vertices and denote the vertices by $V_C := \{c_i\}_{i \in [2|R|+|S|]}$;
   c. V_T of $12|R|$ vertices and denote the vertices by $V_T := \{t_i\}_{i \in [9|R|]}$.
2. Complete (V_S, V_C) to a complete bipartite graph.
3. Complete V_C to a clique, excluding the following edges: $\{(c_{2j-1}, c_{2j}) : j \in [|R|]\}$.
4. $\forall j \in [|R|]$, connect
   a. each vertex in $\{t_{12j-w}\}_{w \in [0,3]}$ to $\{x_j, c_{2j-1}, y_j\}$,
   b. each vertex in $\{t_{12j-w}\}_{w \in [4,7]}$ to $\{c_{2j-1}, y_j, c_{2j}\}$,
   c. each vertex in $\{t_{12j-w}\}_{w \in [8,11]}$ to $\{y_j, c_{2j}, z_j\}$.

   See the following figure for an illustration:



.

Proof of Theorem 4. The betweenness problem (R, S) has a satisfying solution if and only if $G_{R,S}$ is streamable.

Suppose $G_{R,S}$ is a streamable graph, and there is a streaming ordering $\sigma : V(G_{R,S}) \mapsto [|V(G_{R,S})|]$ of $G_{R,S}$. We assert that the sub-ordering of σ induced by V_S is a solution to the betweenness problem (R, S). That is because due to Lemma 7, $\forall j \in [|R|]$,

1. $\{t_{12j-w}\}_{w\in[0,3]}$ enforces $\{x_j, c_{2j-1}, y_j\}$ to be streamed in σ,
2. $\{t_{12j-w}\}_{w\in[4,7]}$ enforces $\{c_{2j-1}, y_j, c_{2j}\}$ to be streamed in σ,
3. $\{t_{12j-w}\}_{w\in[8,11]}$ enforces $\{y_j, c_{2j}, z_j\}$ to be streamed in σ.

In either of the three cases above, the three vertices induce a path and there is only two valid Hamiltonian paths in the induced subgraph, so the three constraints are equivalent to the following:

1. $\sigma(x_j) < \sigma(c_{2j-1}) < \sigma(y_j)$ or $\sigma(y_j) < \sigma(c_{2j-1}) < \sigma(x_j)$,
2. $\sigma(c_{2j-1}) < \sigma(y_j) < \sigma(c_{2j})$ or $\sigma(c_{2j}) < \sigma(y_j) < \sigma(c_{2j-1})$,
3. $\sigma(y_j) < \sigma(c_{2j}) < \sigma(z_j)$ or $\sigma(z_j) < \sigma(c_{2j}) < \sigma(y_j)$.

And it is easy to see that to satisfy the three constraints above, we must have $\sigma(x_j) < \sigma(c_{2j-1}) < \sigma(y_j) < \sigma(c_{2j}) < \sigma(z_j)$ or $\sigma(z_j) < \sigma(c_{2j}) < \sigma(y_j) < \sigma(c_{2j-1}) < \sigma(x_j)$, and in either case y_j lies between x_j and z_j. Therefore, the sub-ordering induced by V_S is a solution to the betweenness problem (R, S).

Now suppose S has an ordering σ that satisfies the betweenness problem, we can construct a streaming ordering of $V(G_{R,S})$. In the following construction, we will use π to denote the constructed sequence of vertices, which will finally grow to a permutation of $V(G_{R,S})$ and the streaming ordering will be π^{-1}. Initially, π is empty.

1. Append all the V_T vertices in arbitrary order to π.
2. Append all the V_S vertices in the order of σ to π.
3. $\forall s \in S$, insert $c_{2|S|+\sigma(s)}$ before s (σ can be replaced with any function that maps S to $[|S|]$).
4. $\forall j \in [|R|]$, find x_j, y_j, z_j in π. If they are in the order of x_j, y_j, z_j, we insert c_{2j-1} after x_j and c_{2j} after y_j; otherwise, insert c_{2j} after z_j and c_{2j-1} after y_j.

Now we verify that for all vertices, the succeeding neighbours are streamed in $\tau := \pi^{-1}$.

1. V_T: $\forall j \in [|R|], w \in [0, 11]$, $N^+_\tau(t_{12j-w}) = N(t_{12j-w})$, and the path is completed by the insertion of c_{2j-1}, c_{2j}.
2. V_S: $s \in V_S$, $N^+_\tau(s)$ is a subset of V_C, which almost induce a clique. We claim, in the sub-ordering induced by $N^+_\tau(s)$, there will not be a pair of c vertices that are adjacent while disconnected, namely, $\nexists\, u, v \in V_C, (v, u) \notin E \wedge \tau|_{N^+_\tau(c)}(u) = \tau|_{N^+_\tau(c)}(v) + 1$. This is because, $\forall j \in [|R|]$, there will be a y_j between (c_{2j-1}, c_{2j}), so $c_{2|S|+\sigma(y_j)}$ will also lie

between these two. Moreover, $c_{2|S|+\sigma(y_j)} \in N^+_\tau(s)$, so the claim is true and $N^+_\tau(s)$ must be streamed in σ.

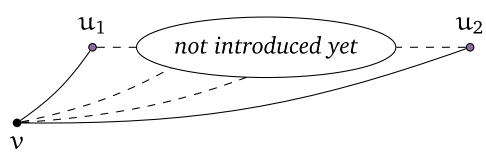
3. V_C: $\forall c \in V_C$, $N^+_\tau(c)$ is the union of a subset of V_C and a subset of V_S. It is streamed because there are no two vertices in V_S that are adjacent in the sub-ordering induced by $N^+_\tau(c)$, namely, $\nexists\, u, v \in V_S, \tau|_{N^+_\tau(c)}(u) = \tau|_{N^+_\tau(c)}(v) + 1$. The argument is similar to the one for V_S.

◀

3 Recognition is fixed-parameter tractable

▶ **Theorem 8.** *The recognition of streamable graphs is fixed-parameter tractable parameterized by the treewidth of the input graph.*

Given a graph of G and associated *nice* tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of width k, we can check whether G is an streamable graph by a bottom-up checking procedure. The non-hereditary property of streamable graphs makes it tricky to compress the solution space, since it might be the case that the subgraph relies on some vertices from upper bags to become streamable, as demonstrated below:



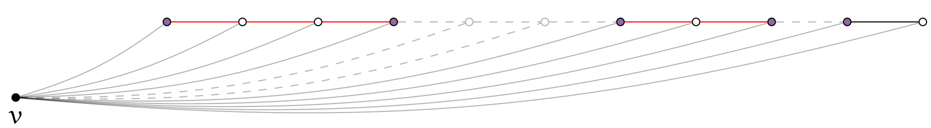
Suppose we finally construct an order π, and we consider $\pi|_{N^+_\pi(v)}$. Then during the construction process, each vertex in the sequence will belong to one of the following three types:

(T1) a vertex not introduced yet,
(T2) a vertex currently in bag, or
(T3) a vertex already forgotten.

One important fact is about the sequence is as follows:

▶ **Lemma 9.** *At any moment, no (T1) vertex will be adjacent to a (T3) vertex.*

Proof. If there exists a (T1) vertex v adjacent to a (T3) vertex u, it means there will be no bag in $\mathcal{T}$ containing both u, v, while (u, v) is an edge in G. Therefore, the tree decomposition is not valid. ◀



Lemma 9 implies if we ignore all (T1) vertices, the sequence of succeeding neighbors of t will be composed of several segments (colored red) of (T2) and (T3) vertices, with each segment starting and ending with (T2) vertices, except that the first segment may start with a (T1) vertex, and the last segment may end with a (T1) vertex. Formally,

▶ **Definition 10** (Valid order). *For some node $t \in T$, we say a order $\pi \in \mathrm{Sym}(V_t)$ is valid for t if and only if it satisfies the following conditions.*

(V1) *For each $v \in X_t$, if we look at $s = \pi|_{N^+_\pi(v)}$, then two adjacent vertices s_i, s_{i+1} are not connected only if $s_i, s_{i+1} \in X_t$.*

(V2) *For each $v \in V_t \setminus X_t$, $N_\pi^+(v)$ is streamed in π.*
The first condition allows us to keep track of $\pi|_{N_\pi^+(v)}$ only using the endpoints in $X_t \cup \{\perp, \perp'\}$, where $\perp, \perp'$ are two virtual isolated vertices. When constructing a valid order, we may insert a vertex into it. We have the following corollary of Lemma 9.

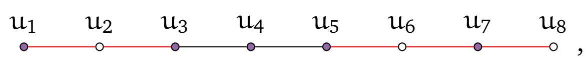
▶ **Corollary 11.** *If a new vertex u is inserted to a valid order π for t, the result order is valid only if $\forall v \in X_t$, the predecessor and successor of u in $\pi|_{N_\pi^+(v)}$ are (T2) vertices.*

From Corollary 11, we see that only the endpoints of (T2) points will be useful for the construction of a valid order. Therefore, all information we need about a valid order π can be encompassed by a canonical representation defined as follows.

▶ **Definition 12 (Canonical representation of a valid order).** *Given a node $t \in V(T)$ and an order $\pi \in \mathrm{Sym}(V_t)$ and $v \in X_t$, we define the canonical representation of $\pi|_{N_\pi^+(v)}$ in π under X_t as an ordered set of endpoints $\{(p_1, q_1), (p_2, q_2), \ldots, (p_k, q_k)\}$ such that for all i,*

1. *$p_i, q_i \in X_t \cup \{\perp, \perp'\}$, only p_1 is allowed to be $\perp$, only q_k is allowed to be $\perp'$,*
2. *segments do not intersect, i.e., $\pi^{-1}(p_i) < \pi^{-1}(q_i) < \pi^{-1}(p_{i+1})$,*
3. *the subset of elements in $N_\pi^+(v)$ covered by this segment, i.e., $C_i = \{v \mid v \in N_\pi^+(v) \wedge \pi^{-1}(p_i) \leqslant \pi^{-1}(v) \leqslant \pi^{-1}(q_i)\}$, is streamed in π^{-1},*
4. *there are no two consecutive X_t elements in $\pi|_{C_i}$, and*
5. *all segments together cover all $N_\pi^+(v)$, i.e., $\bigcup_{i=1}^k C_i \setminus \{\perp, \perp'\} = N_\pi^+(v)$.*

For example, for the following graph where empty vertices belong to $N_\pi^+(v) \setminus X_t$ and filled vertices belong to X_t



the canonical representation is $\{(u_1, u_3), (u_4, u_4), (u_5, \perp')\}$. Given $t, v \in X_t, \pi \in \mathrm{Sym}(V_t)$, we can compute the canonical representation of $N_\pi^+(v)$ under X_t in linear time. Moreover, the representation is unique.

In our algorithm, when working from the leaves to the root, we will keep track of all canonical representations of $\pi|_{N_\pi^+(v)}$ for $v \in X_t$. Now we formally describe the algorithm.

▶ **Definition 13.** *Given a sequence τ, we define $\mathcal{E}_\tau$ to be set of all possible canonical representation with endpoints in τ, namely*

$$\mathcal{E}_\tau = \{\{(\tau_{l_1}, \tau_{r_1}), (\tau_{l_2}, \tau_{r_2}), \ldots, (\tau_{l_k}, \tau_{r_k})\} \mid 0 \leqslant l_1 \leqslant r_1 < l_2 \leqslant r_2 < \cdots < l_k \leqslant r_k \leqslant |\tau|+1\}.$$

We define $\tau_0 = \perp, \tau_{|\tau|+1} = \perp'$.

For each node $t \in V(T)$ we define c_t to be the set of all $(\tau, b) \in \mathrm{Sym}(X_t) \times (X_t \times \mathcal{E}_\tau)$ such that there exists a *valid* $\pi \in \mathrm{Sym}(V_t)$ satisfying the following conditions.
(C1) τ is the restriction of π to X_t.
(C2) For each $v \in X_t$, b_v is the canonical representation of $\pi|_{N_\pi^+(v)}$ under X_t.
(C3) For each $v \in V_t \setminus X_t$, $N_\pi^+(v)$ is streamed in π^{-1}.
Now we discuss how c can be constructed in a bottom-up manner, and finally the graph is streamable if and only if for the root node r of $V(T)$, $c_r \neq \emptyset$.

Transitions

Leaf node: If t is a leaf node, then $c_t = \{(\emptyset, \emptyset)\}$.
Introduce node: If t is an introduce node with child t' such that $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$, then construct c_t as follows.

for $(\tau', b') \in c_{t'}$ **do**
  for $j \in \{0, 1, \ldots, \ell\}$ **do**
    $\tau \leftarrow (\tau'_1, \ldots, \tau'_j, v, \tau'_{j+1}, \ldots, \tau'_\ell)$;
    $b \leftarrow b' \cup (v, \{(p, p) \mid p \in N^+_\tau(v)\})$;
    for $u \in N^-_\tau(v)$ **do**
      if *v intersects with a segment in b_u* ***or***
        the segment preceding v in b_u ends with $\perp'$ ***or***
        the segment following v in b_u starts with $\perp$ **then**
        $b_u \leftarrow \Box$;
      else
        $b_u \leftarrow b_u \cup \{(v, v)\}$;
    if $\forall u \in X_t, b_u \neq \Box$ **then**
      $c_t \leftarrow c_t \cup \{(\tau, b)\}$;

Forget node: If t is a forget node with child t' such that $X_t = X_{t'} \setminus \{v\}$ for some $v \in X_{t'}$, then construct c_t as follows:

for $(\tau', b') \in c_{t'}$ **do**
  if $\exists$ *adjacent* $(p, q), (p', q') \in b'_v$ *while* $q \notin N(p')$ **then**
    continue;
  $(\tau, b) \leftarrow (\tau'|_{X_t}, b'|_{X_t})$;
  for $u \in X_t \cup N(v)$ *s.t.* $\tau'^{-1}(u) < \tau'^{-1}(v)$ **do**
    if $\exists (v, q) \in b_u$ **then**
      if *there is no segment preceding* (v, q) *in* b_u **then**
        $b_u \leftarrow b_u \setminus \{(v, q)\} \cup \{(\perp, q)\}$;
      else
        $(p', q') \leftarrow$ the segment preceding (v, q) in b_u;
        if q' *is connected to* v **then**
          $b_u \leftarrow b_u \setminus \{(p', q'), (v, q)\} \cup \{(p', q)\}$;
        else
          $b_u \leftarrow \Box$;
    if $\exists (p, v) \in b_u$ **then**
      Similar to the above case;
  if $\forall u \in X_t, b_u \neq \Box$ **then**
    $c_t \leftarrow c_t \cup \{(\tau, b)\}$;

Join node: If t is a join node with children t' and t'', then we construct c_t as follows:

for $((\tau', b'), (\tau'', b'')) \in c'_t \times c''_t$ *s.t.* $\tau' = \tau''$ **do**
  for $u \in X_t$ **do**
    if $\exists (p', q') \in b'_u, (p'', q'') \in b''_u$ *s.t.* $|(p', q') \cap (p'', q'')| > 1$ **then**
      $b_u \leftarrow \Box$;
    else
      $b_u \leftarrow b'_u \cup b''_u$ with unit segments that are covered deleted;
  if $\forall u \in X_t, b_u \neq \Box$ **then**
    $c_t \leftarrow c_t \cup \{(\tau', b)\}$;

Now we prove the correctness of the algorithm. Namely, we prove:

▶ **Lemma 14.** *$\mathcal{F}(t, \tau, b) \neq \emptyset$ if and only if there exists a transition in the algorithm above that inserts (τ, b) to c_t.*

Proof. We elaborate the lemma in two directions.

- *If*: Starting from a (τ', b') with $\mathcal{F}(t', \tau', b') \neq \emptyset$, if we successfully construct a (τ, b) following the transition rules, then $\mathcal{F}(t, \tau, b) \neq \emptyset$.
- *Only if*: Suppose $\mathcal{F}(t, \tau, b) \neq \emptyset$. Then there is a (τ', b') with $\mathcal{F}(t', \tau', b') \neq \emptyset$ from which we can construct (τ, b) and add it c_t following the transition rules.

Now consider the four types of nodes separately. The case of leaf node is trivial.

Introduce node: *If*: Suppose there is $\pi' \in \mathcal{F}(t', \tau', b')$, and we enumerate the position of v in τ and define π to be the order that inserts v to arbitrary position in π' as long as $\tau = \pi|_{X_t}$ is satisfied. Then for this π to be valid, we need to ensure (V1) in Definition 10. For the newly inserted v, $N_\pi^+(v) \subset X_t$, so (V1) trivially holds. For $u \in N_\tau^-(v)$, we need to ensure if we look at $N_\pi^+(u)$, the predecessor (if there is one) and the successor (if there is one) of v must belong to $X_{t'}$. This condition can be equivalently captured in terms of b'_u as follows:

1. v must not intersect any existing segments in b'_u, and
2. (v, v) cannot succeed a $(\perp, \cdot)$ segment or precede a $(\cdot, \perp')$ segment.

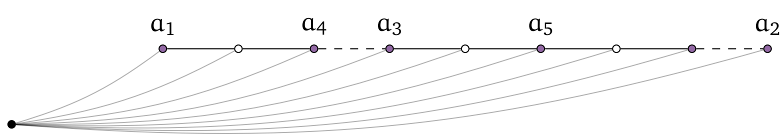
If the check above passes, then π will be valid, b constructed as

1. b_v consists of all unit segments in $\pi|_{N_\tau^+(v)}$,
2. for all $u \in N_\tau^-(v)$, $b_u \leftarrow b'_u \cup \{(v, v)\}$, and
3. for the remaining u, $b_u \leftarrow b'_u$,

will be the set of canonical representations of $\pi|_{N_\pi^+(\cdot)}$ under X_t. Therefore, if b is successfully constructed in the algorithm, we can construct a $\pi \in \mathcal{F}(t, \tau, b)$.

Only if: Suppose there is a $\pi \in \mathcal{F}(t, \tau, b)$. Then we define $\pi' = \pi|_{X_{t'}}$, and it is easy to see that π' is valid for t', since v is not connected to any forgotten vertices, so (C2) is unaffected if we delete v. Moreover, not connecting to forgotten vertices means v will appear as a unit segment in b_u. Then we can set $\tau' = \pi'|_{X_{t'}}, b'_u = b_u \setminus \{(v, v)\}$ for all $u \in X_{t'}$, then $\pi' \in \mathcal{F}(t', \tau', b')$. It is easy to see that this (τ', b') will transit to (τ, b) and insert (τ, b) to c_t.

Forget node: *If*: Suppose there is a $\pi' \in \mathcal{F}(t', \tau', b')$. For π' to remain valid for t, we need the following checks. We first check whether the succeeding neighbors of v are streamed by checking whether the endpoints of adjacent segments are connected, as shown below (where we check the existence of dashed edges):



If the check is passed, then (V2) is satisfied. Now we consider how to ensure (V1) for $N_{\tau'}^-(v)$. For v in $\pi|_{N_{\pi'}^+(u)}$, there will be several possibilities:

1. v is the first vertex in the sequence (a_1),
2. v is the last vertex in the sequence (a_2),
3. v succeeds (T2) vertex (a_3),
4. v precedes (T2) vertex (a_4), or
5. v is adjacent to two (T3) vertices (a_5).

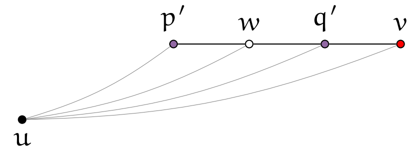
We only discuss cases 1, 3 and 5. For case 1, it is safe to delete v, but as a consequence, v must remain the first vertex in $\pi|_{N_\pi^+(u)}$ for any future π, since if a new vertex is inserted in front of v, it can never be connected to v. For case 3, suppose the predecessor of v is q', then for (V1) to hold, we must ensure q' is connected to v. For case 5, it is safe to delete v with no further consequence. Then these conditions can also be alternatively captured using the canonical representation b'_u.

1. For case 1, v is the left endpoint of the first segment, and to make sure no vertex can be inserted in front of v, we need to replace v with $\perp$ in the representation.

2. For case 3, suppose the segment containing v is (v, q), then we need to look at the predecessor (p', q') of (v, q) in b'_u, and make sure q' and v are connected.
3. For case 5, v will not appear in b'_u, so nothing needs to be done.

If the check above passes, then π' will be valid for t. Now we explain the construction of b on top of b'. We only elaborate for case 3.

1. If $p' \neq q'$, the neighborhood of q' will look like the following:



   Then by Corollary 11, the predecessor and successor of q' must be fixed to be w, v respectively in the future construction process. As a result, we can actually *forget* q' in advance, since it is *sealed* by w and v will never be utilized in the future.
2. If $p' = q'$, it is natural to see we should absorb the segment (p', q') into (v, q).

Therefore, in either case, we should replace $(p', q'), (v, q)$ with a concatenated segment (p', q).

Note that we only show how to deal with the checks and modifications considering the predecessor of v. A similar process should be done again for the successor of v to make sure π' is valid and calculate the final b, which will be the set of canonical representations of $\pi|_{N^+_{\pi'}(\cdot)}$ under X_t. Therefore, if b is successfully constructed in the algorithm, we can construct a $\pi' \in \mathcal{F}(t, \tau, b)$.

Only if: Suppose there is a valid $\pi \in \mathcal{F}(t, \tau, b)$, we first remark π is also valid for t'. Then we simply evaluate b' using π on bag t', and we will have $\pi \in \mathcal{F}(t', \tau', b')$. This (τ', b') will transit to some (τ, b'') and insert (τ, b'') to c_t. However, since π is fixed, the set of canonical representations is unique, so $b = b''$.

Join node: *If*: Suppose there is $\pi' \in \mathcal{F}(t', \tau', b'), \pi'' \in \mathcal{F}(t'', \tau'', b'')$ with $\pi'|_{X_t} = \pi''|_{X_t}$, and we consider how these two orders can be merged into one valid order for V_t. Specifically, we need to be careful with the succeeding neighbors of $u \in X_t$ and make sure the order is valid.

Note that the full order π must not keep the relative position among vertices in X_t, V'_t, V''_t. Now we give the following remark:

▶ **Remark 15.** Define $\zeta = \pi|_{N^+_\pi(u) \cap X_t}$. Then if we look at $\pi|_{N^+_\pi(u)}$, between any pair of ζ_i, ζ_{i+1}, vertices from $V_{t'}$ and vertices from $V_{t''}$ cannot both appear.

Proof: Suppose the remark is incorrect, then there will be alternative appearance of $V_{t'}$ vertices and $V_{t''}$ vertices between ζ_i, ζ_{i+1}, but these two sets of vertices are not connected, so (V1) is violated.

Therefore, for a valid π to exist, π', π'' must satisfy:

▶ **Remark 16.** ζ_i is adjacent to ζ_{i+1} in either $\pi'|_{N^+_{\pi'}(u)}$ or $\pi''|_{N^+_{\pi''}(u)}$.

On the other hand, given Remark 16, then if we write $\pi' = s'_0 \circ \langle \tau'_1 \rangle \circ s'_1 \circ \langle \tau'_2 \rangle \circ \ldots \circ \langle \tau'_{|\tau'|} \rangle \circ s'_{|\tau'|}$ and $\pi'' = s''_0 \circ \langle \tau''_1 \rangle \circ s''_1 \circ \langle \tau''_2 \rangle \circ \ldots \circ \langle \tau''_{|\tau''|} \rangle \circ s''_{|\tau''|}$ (where $\circ$ denotes sequence concatenation, and $\langle a \rangle$ denotes the sequence of a single element a), then we can simply construct a valid π as

$$s'_0 \circ s''_0 \circ \langle \tau_1 \rangle \circ s'_1 \circ s''_1 \circ \langle \tau_2 \rangle \circ \ldots \circ \langle \tau_{|\tau|} \rangle \circ s'_{|\tau|} \circ s''_{|\tau|}.$$

Therefore Remark 16 is necessary and sufficient, which can also be equivalently captured using b'_u, b''_u. In a canonical representation, only non-unit segments contains forgotten vertices. So the condition given by Remark 16 is equivalent to any non-unit segments in b'_u and b''_u does not intersect (except for the endpoints). If the check passes, then the canonical representation b_u for $N^+_\pi(u)$ can be obtained by uniting b'_u and b''_u and remove the unit segments that are covered by some other segments, concluding $\pi \in \mathcal{F}(t, \tau, b)$.

Only if: Suppose there is a valid $\pi \in \mathcal{F}(t, \tau, b)$. Then similar argument suggests that for all $u \in X_t$, each segment in b_u either only contain vertices from $V_{t'}$ or vertices from $V_{t''}$, and we can partition all non-unit segments in b_u based on this condition, and properly completing two parts with unit segments will give us b'_u, b''_u. Then we define $\pi' = \pi|_{V'_t}, \pi'' = \pi|_{V''_t}$, and b', b'' will be the set of canonical representations of $\pi'|_{N^+_{\pi'}(\cdot)}, \pi''|_{N^+_{\pi''}(\cdot)}$, namely we have $\pi' \in \mathcal{F}(t', \tau', b'), \pi'' \in \mathcal{F}(t'', \tau'', b'')$. It is clear that this $(\tau', b'), (\tau'', b'')$ will transit to (τ, b) and insert (τ, b) to c_t.

◀

Given the correctness of the algorithm, we can finally prove the main theorem of this section.

Proof of Theorem 8. For all $t \in V(T)$, the number of different (τ, b) is bounded by a function of the tree width k, and therefore, the time complexity of transition is also bounded by a function of k, say $f(k)$. Since a nice tree decomposition consists of kn bags, the total time complexity is bounded by $kf(k) \cdot n$. Therefore, the recognition of streamable graphs is fixed-parameter tractable with respect to the tree width of the input graph. ◀

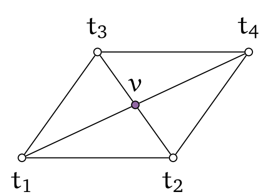
4 Search-to-decision reduction

▶ **Theorem 17.** *Suppose we have an oracle $\mathcal{O}_s$ that determines whether a graph G is streamable. Then, for a streamable graph G, there exists a polynomial-time algorithm that produces a streaming ordering of G using a polynomial number of queries to $\mathcal{O}_s$.*

The idea is to reconstruct a streaming ordering step-by-step, from the last vertex to the first. Specifically, we will use the oracle to find a vertex v such that there is a streaming ordering that ends with v, and then find a vertex w such that there is a streaming ordering that ends with (w, v), and repeat the process until the ordering is complete.

Before we proceed, we need the following lemma:

▶ **Lemma 18.** *In any streaming ordering $\sigma : \{v\} \cup \{t_b\}_{b \in [4]} \mapsto [5]$ of the following graph*



,

we must have $\sigma^{-1}(5) \in \{t_b\}_{b \in [4]}$, i.e., the ordering must not end with v.

Proof. Suppose there is a streaming ordering σ with $\sigma^{-1}(5) = v$, which means σ starts with a prefix which is a permutation of $\{t_b\}_{b \in [4]}$. Since the σ is a streaming ordering, the prefix should be a streaming ordering for the subgraph induced by $\{t_b\}_{b \in [4]}$, but that induced subgraph is a cycle and no way an streamable graph. Thus we have a contradiction. ◀

We will make use of the lemma above to achieve the following: given a sequence of vertices $w_1, w_2, \ldots, w_\ell$ determine if there is a streaming ordering σ with the sequence as a suffix, i.e., $\sigma(w_i) = n - \ell + i$ for all $i \in [\ell]$. Now we elaborate our construction.

Given G and $w = (w_1, w_2, \ldots, w_\ell)$, we create $G' := c(G, w)$ as follows:

1. Let $G' := G$.

2. $\forall i \in [\ell]$, create 4 new vertices $t_{i,1}, t_{i,2}, t_{i,3}, t_{i,4}$, and add edges $\{(w_i, t_{i,b}) : b \in [4]\} \cup \{(t_{i,b}, t_{i,b \bmod 4+1}) : b \in [4]\}$ to G'.
3. $\forall i, j \in [\ell]$ with $i < j$, if $(w_i, w_j) \in E$, then add edge $(t_{i,1}, w_j)$ to G'.

Then we have the following lemma:

▶ **Lemma 19.** *In a streaming order σ' for G', $\forall 1 \leqslant i < j \leqslant \ell, (w_i, w_j) \in E$, we have $\sigma'(w_i) < \sigma'(t_{i,1}) < \sigma'(w_j)$.*

Proof. We mention the following remark:

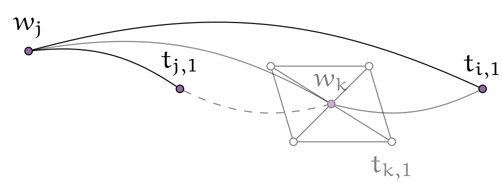
▶ Remark 20. $\forall i \in [\ell]$, if $\exists v \in V'$ with $(w_i, v) \in E'$ and $\sigma'(w_i) < \sigma'(v)$, then

$$\max(\sigma'(t_{i,b}) : b \in \{2,3,4\}) < \sigma'(u_i) < \sigma'(t_{i,1}). \tag{3}$$

Now we first prove the second inequality by contradiction. Namely, we prove

$$\forall 1 \leqslant i < j \leqslant \ell \text{ with } (w_i, w_j) \in E, \sigma'(t_{i,1}) < \sigma'(w_j). \tag{4}$$

Suppose $\exists i, j, i < j$ with $(w_i, w_j) \in E, \sigma'(t_{i,1}) > \sigma'(w_j)$, we find such a pair (i, j) with the largest $\sigma'(w_j)$. Recall that $(t_{i,1}, w_j) \in E'$ since $i < j$. Then by Remark 20, we have $\sigma'(w_j) < \min\{\sigma'(t_{j,1}), \sigma'(t_{i,1})\}$. Without loss of generality, we assume $\sigma'(t_{j,1}) < \sigma'(t_{i,1})$. Then for the succeeding neighbors of w_j to be streamed, there must be a path of neighbors of w_j from $t_{j,1}$ to $t_{i,1}$. We consider the second last vertex on this path, and denote it by w_k, looking as follows:

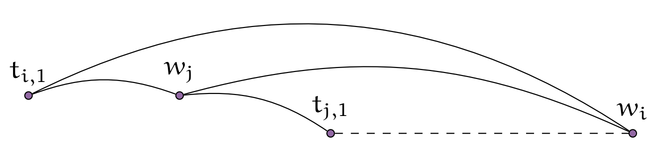


.

Note that since $(w_k, t_{i,1}) \in E'$, we know $i < k$. Still, due to Remark 20, we have $\sigma'(w_k) < \min\{\sigma'(t_{k,1}), \sigma'(t_{i,1})\}$, giving us another pair (i, k) with $\sigma'(w_k) > \sigma'(t_{j,1}) > \sigma'(w_j)$, contradicting the assumption of (i, j) being the pair with the largest $\sigma'(w_j)$. Therefore, there will not be such a pair (i, j) and Equation (4) must hold. Now it remains to show

$$\forall 1 \leqslant i < j \leqslant \ell \text{ with } (w_i, w_j) \in E, \sigma'(w_i) < \sigma'(w_j). \tag{5}$$

It is easy to see that Equations (4) and (5) together with Remark 20 prove the lemma. Suppose $\exists i, j, i < j$ with $(w_i, w_j) \in E$ and $\sigma'(w_i) > \sigma'(w_j)$, then by Remark 20, $\sigma'(w_j) < \sigma'(t_{j,1})$. Similarly, for the succeeding neighbors of w_j to be streamed, there must exist a path of neighbors of w_j from $t_{j,1}$ to w_i. However, by Equation (4), $\sigma'(t_{i,1}) < \sigma'(w_j)$ since $i < j$, then by Remark 20 w_i has no succeeding neighbors. This means it can only appear on the right end of the aforementioned path, giving us $\sigma'(t_{j,1}) < \sigma'(w_i)$, as shown below:



.

Now we select a pair (i, j) that $(w_i, w_j) \in E$ and $\sigma'(w_i) > \sigma'(w_j)$ with the smallest $\sigma'(w_i) - \sigma'(w_j)$, and look at the second last vertex on this path. There are two different cases:

1. $t_{k,1}$ for some k: then since $(t_{k,1}, w_i) \in E'$, we know $k < i < j$, giving us a pair (k, j) violating (4), thus impossible.
2. w_k for some k: then we know $j < k$ since otherwise (k, j) would be a pair with smaller $\sigma'(w_k) - \sigma'(w_j)$. However, this gives us $i < j < k$, meaning that (i, k) is a pair with smaller $\sigma'(w_k) - \sigma'(w_i)$, contradicting the assumption.

Thus we finalize the proof of (5), together with (4) we complete the proof of Lemma 19. ◀

▶ **Lemma 21.** *In a streaming order σ' for G', $\forall v \in V \setminus w, i \in [\ell]$ with $(v, w_i) \in E$, we have $\sigma'(v) < \sigma'(w_i)$.*

Proof. Suppose there is such (v, i) with $\sigma'(w_i) < \sigma'(v)$, we then find a pair with the smallest $\sigma'(v) - \sigma'(w_i)$. In this case, if we look at the path $N^+_{\sigma'}(w_i)$, it is easy to see that all $N^+_{\sigma'}(w_i) \cap w$ will appear after v in this path. However, by Remark 20, $t_{i,1} \in N^+_{\sigma'}(w_i)$, and by Lemma 19, $t_{i,1}$ appear before all $N^+_{\sigma'}(w_i) \cap w$. In this case, there cannot be a path between $t_{i,1}$ and v, and thus there will be no such (v, i). ◀

The lemma above says that all w vertices should appear after $V \setminus w$ vertices (if they are connected). Lemma 19 and Lemma 21 together give us enough structural information to determine if there is a streaming ordering with w as a suffix. Finally, we have the following lemma:

▶ **Lemma 22.** *Given G, w, there is a streaming ordering σ of G with w as a suffix if and only if G' is streamable.*

Proof. Suppose G' is streamable with a streaming ordering σ', and we use $\sigma_{V\setminus w} := \sigma'|_{V\setminus w}$ to denote the restriction of σ' to $V \setminus w$. Then a streaming ordering σ of G with w as a suffix can be defined as follows:

$$\sigma^{-1} = \left(\sigma^{-1}_{V\setminus w}(1), \sigma^{-1}_{V\setminus w}(2), \ldots, \sigma^{-1}_{V\setminus w}(n-k), w_1, w_2, \ldots, w_k\right). \tag{6}$$

To see σ is a streaming ordering, we consider all vertices in V and argue their succeeding neighbors are streamed in σ. For $u \in V$, we denote the succeeding neighbors of u in (G, σ) by $N^+_\sigma(u)$ and the succeeding neighbors of u in (G', σ') by $N^{+\prime}_{\sigma'}(u)$. We first mention the following corollary which is a consequence of Lemma 19 and Lemma 21.

▶ **Corollary 23.** *$\forall(u, v) \in E' \cap V \times V$, if $\sigma'(u) < \sigma'(v)$, then $\sigma(u) < \sigma(v)$.*

Then we consider the following two cases:

1. $\forall v \in V \setminus w$: $N^{+\prime}_{\sigma'}(v) = N^+_\sigma(v)$. Then by Corollary 23, if the first set is streamed in σ', then the second set is streamed in σ.
2. $\forall i \in [\ell], w_i$: $N^{+\prime}_{\sigma'}(w_i) = N^+_\sigma(w_i) \cup \{t_{i,1}\}$. Moreover, by Lemma 19, $\sigma'(t_{i,1}) < \min_{u \in N^+_\sigma(w_i)} \sigma'(u)$, meaning that $N^+_\sigma(w_i)$ can be streamed in σ' without $t_{i,1}$, thus by Corollary 23 $N^+_\sigma(w_i)$ is streamed in σ.

Conversely, suppose there is a streaming ordering σ of G with w as a suffix, then there is a streaming ordering σ' of G' defined as follows:

$$\begin{aligned}\sigma'^{-1} = (&\sigma^{-1}(1), \sigma^{-1}(2), \ldots, \sigma^{-1}(n-k),\\ &t_{1,2}, t_{1,3}, t_{1,4}, w_1, t_{1,1},\\ &t_{2,2}, t_{2,3}, t_{2,4}, w_2, t_{2,1},\\ &\ldots,\\ &t_{k,2}, t_{k,3}, t_{k,4}, w_k, t_{k,1}),\end{aligned} \tag{7}$$

and the lemma is proved. ◀

Based on Lemma 22, we can give the algorithm to find a streaming ordering of G in Algorithm 1. Clearly, the algorithm terminates in polynomial time and makes $O(n^2)$ queries to $\mathcal{O}_s$.

```
Function CHECK-SUFFIX(G, w):
    G′ ← c(G, w);
    return 𝒪_s(G′);
Function STREAMING-ORDERING(G):
    if not 𝒪_s(G) then
        return ∅;
    W ← [];
    for i ← 1, 2, . . . , n do
        for u ∈ V(G) \ w do
            if CHECK-SUFFIX(G, [u] + w) then
                W ← [u] + w;
                break;
    return W^{−1};
```

Algorithm 1 Finding a streaming ordering of G.

References

1 J Opatrny. Total Ordering Problem. *SIAM Journal on Computing*, 8(1):111–114, 1979. doi:10.1137/0208008.