Characterizations and Directed Path-Width of Sequence Digraphs

Frank Gurski\textsuperscript{1} · Carolin Rehs\textsuperscript{1} · Jochen Rethmann\textsuperscript{2}

Accepted: 3 October 2022 / Published online: 23 November 2022 © The Author(s) 2022

Abstract
Computing the directed path-width of a directed graph is an NP-hard problem. Even for digraphs of maximum semi-degree 3 the problem remains hard. We propose a decomposition of an input digraph $G = (V, A)$ by a number $k$ of sequences with entries from $V$, such that $(u, v) \in A$ if and only if in one of the sequences there is an occurrence of $u$ appearing before an occurrence of $v$. We present several graph theoretical properties of these digraphs. Among these we give forbidden subdigraphs of digraphs which can be defined by $k = 1$ sequence, which is a subclass of semi-complete digraphs. Given the decomposition of digraph $G$, we show an algorithm which computes the directed path-width of $G$ in time $O(k \cdot (1 + N)^k)$, where $N$ denotes the maximum sequence length. This leads to an XP-algorithm w.r.t. $k$ for the directed path-width problem. Our result improves the algorithms of Kitsunai et al. for digraphs of large directed path-width which can be decomposed by a small number of sequences and confirm their conjecture that semicompleteness is a useful restriction when considering digraphs.

Keywords Digraphs · Directed path-width · Transitive tournaments · XP-algorithm

Mathematics Subject Classification (2010) 05C85 · 90C39

A short version of this paper appeared in the Proceedings of the International Conference on Combinatorial Optimization and Applications (COCOA 2018) [1].

Frank Gurski
frank.gurski@hhu.de

Carolin Rehs
carolin.rehs@hhu.de

Jochen Rethmann
jochen.rethmann@hsnr.de

\textsuperscript{1} University of Düsseldorf, Institute of Computer Science, Algorithmics for Hard Problems Group, 40225, Düsseldorf, Germany

\textsuperscript{2} Faculty of Electrical Engineering and Computer Science, Niederrhein University of Applied Sciences, 47805 Krefeld, Germany
1 Introduction

Let $Q = \{q_1, \ldots, q_k\}$ be a set of $k$ sequences $q_i = (b_{i,1}, \ldots, b_{i,n_i})$, $1 \leq i \leq k$. All $n_1 + \ldots + n_k$ items $b_{i,j}$ are pairwise distinct. Further there is a function $t$ which assigns every item $b_{i,j}$ with a type $t(b_{i,j})$. The sequence digraph $g(Q) = (V, A)$ for a set $Q = \{q_1, \ldots, q_k\}$ has a vertex for every type and an arc $(u, v) \in A$ if and only if there is some sequence $q_i$ in $Q$ where an item of type $u$ is on the left of some item of type $v$.\footnote{A related but different concept for undirected graphs is the notation of word-representable graphs. A graph $G = (V, E)$ is word-representable if there exists a word $w$ over the alphabet $V$ such that letters $x$ and $y$ alternate in $w$ if and only if $\{x,y\} \in E$, see [2] for a survey.} The contributions of this paper concern graph theoretic properties of sequence digraphs and an algorithm to compute from a given set $Q$ on $k$ sequences the directed path-width of $g(Q)$. In Section 2 we give preliminaries for graphs and digraphs. In Section 3 we show how to define digraphs from sets of sequences, and vice versa. Further we give methods in order to compute the sequence digraph and also its complement digraph. In Section 4 we consider graph theoretic properties of sequence digraphs. Therefore in Section 4.1 we introduce the class $S_{k,\ell}$ as the set of all sequence digraphs defined by sets $Q$ on at most $k$ sequences that together contain at most $\ell$ items of each type. We show inclusions of these classes and necessary conditions for digraphs to be in $S_{k,\ell}$. In Section 4.2 we give finite sets of forbidden induced subgraphs for all classes $S_{1,1}$ and $S_{1,2}$. It turns out that $S_{1,1}$ is equal to the well known class of transitive tournaments. Since only the first and the last item of each type in every $q_i \in Q$ are important for the arcs in the corresponding digraph all classes $S_{1,\ell}$, $\ell \geq 2$ are equal and can be characterized by one set of four forbidden induced subdigraphs. These characterizations lead to polynomial time recognition algorithms for the corresponding graph classes. Furthermore we give characterizations in terms of special tournaments and conditions for the complement digraph. In Section 4.3 we consider the closure of the classes $S_{k,\ell}$ with respect to the operations taking the converse digraph and taking the complement digraph. In Section 4.4 we characterize sequences whose defined graphs can be obtained by the union of the graphs for the subsequences which consider only the first or last item of each type.

In Section 5 we consider the directed path-width problem on sequence digraphs. We show that for digraphs defined by $k = 1$ sequence the directed path-width can be computed in polynomial time. Further we show that for sets $Q$ of sequences of bounded length, of bounded distribution of the items of every type onto the sequences, or bounded number of items of every type computing the directed path-width of $g(Q)$ is NP-hard. We also introduce a method which computes from a given set $Q$ on $k$ sequences the directed path-width of $g(Q)$, as well as a directed path-decomposition can be computed in time $O(k \cdot (1 + \max_{1 \leq i \leq k} n_i)^k)$. The main idea is to discover an optimal directed path-decomposition by scanning the $k$ sequences left-to-right and keeping in a state the numbers of scanned items of every sequence and the number of active types.

From a parameterized point of view our method leads to an XP-algorithm w.r.t. parameter $k$. While the existence of FPT-algorithms for computing directed path-width is open up to now, there are XP-algorithms for the directed path-width...
Table 1 Values of parameters within XP-algorithms for directed path-width

| Digraphs $G$                          | $d$-pw$(G)$ | $\tau(\text{und}(G))$ | $k$ | $\ell$ |
|--------------------------------------|-------------|------------------------|-----|--------|
| Transitive tournaments               | 0           | $n - 1$                | 1   | 1      |
| Disjoint union of $k'$ transitive tournaments | 0           | $\left(\sum_{i=1}^{k'} n_i\right) - k'$ | $k'$ | 1      |
| Bidirectional complete digraphs $\overrightarrow{K}_n$ | $n - 1$ | $n - 1$ | 1 | 2 |
| Semicomplete $\{\overrightarrow{C}_3, D_0, D_4\}$-free | $[0, n - 1]$ | $n - 1$ | 1 | 2 |
| Semicomplete $\{\text{co-}(\overrightarrow{P}_2), \overrightarrow{C}_3, D_4\}$-free | $[0, n - 1]$ | $n - 1$ | 1 | 2 |
| Disjoint union of $k'$ semicomplete digraphs $\{\text{co-}(\overrightarrow{P}_2), \overrightarrow{C}_3, D_4\}$-free | $[0, n - 1]$ | $\left(\sum_{i=1}^{k'} n_i\right) - k'$ | $k'$ | $2k'$ |

problem for some digraph $G = (V, A)$. The directed path-width can be computed in time $O(|A| \cdot |V|^{2\text{d-pw}(G)/d} - 1)$ by [3] and in time $O(d\text{-pw}(G) \cdot |A| \cdot |V|^{2\text{d-pw}(G)})$ by [4]. Further in [5] it is shown how to decide whether the directed path-width of an $\ell$-semicomplete digraph is at most $w$ in time $(\ell + 2w + 1)^{2w} \cdot |V|^{O(1)}$. All these algorithms are exponential in the directed path-width of the input digraph while our algorithm is exponential within the number of sequences. Thus our result improves theses algorithms for digraphs of large directed path-width which can be decomposed by a small number of sequences (see Table 1 for examples). Furthermore the directed path-width can be computed in time $3^{\tau(\text{und}(G))} \cdot |V|^{O(1)}$, where $\tau(\text{und}(G))$ denotes the vertex cover number of the underlying undirected graph of $G$, by [6]. Thus our result improves this algorithm for digraphs of large $\tau(\text{und}(G))$ which can be decomposed by a small number of sequences (see Table 1 for examples).

Since every single sequence $q_i$ defines a semicomplete $\{\text{co-}(\overrightarrow{P}_2), \overrightarrow{C}_3, D_4\}$-free digraph (cf. Table 2 for the digraphs), we consider digraphs which can be obtained by the union of $k$ special semicomplete digraphs which confirms the conjecture of [5] that semicompleteness is a useful restriction when considering digraphs. In Section 6 we give conclusions and point out open questions.

---

2When considering the directed path-width of almost semicomplete digraphs in [5] the class of semicomplete digraphs was suggested to be “a promising stage for pursuing digraph analogues of the splendid outcomes, direct and indirect, from the Graph Minors project”.

---

© Springer
Table 2 Special digraphs

\[ \begin{array}{cccc}
D_6 & 2\overrightarrow{P}_2 & co-(2\overrightarrow{P}_2) \\
\end{array} \]

2 Preliminaries

We use the notations of Bang-Jensen and Gutin [7] for graphs and digraphs.

**Undirected Graphs** We work with finite undirected graphs \( G = (V, E) \), where \( V \) is a finite set of vertices and \( E \subseteq \{(u, v) \mid u, v \in V, u \neq v\} \) is a finite set of edges. For a vertex \( v \in V \) we denote by \( N_G(v) \) the set of all vertices which are adjacent to \( v \) in \( G \), i.e. \( N_G(v) = \{w \in V \mid \{v, w\} \in E\} \). Set \( N_G(v) \) is called the set of all neighbors of \( v \) in \( G \) or neighborhood of \( v \) in \( G \). The degree of a vertex \( v \in V \), denoted by \( \deg_G(v) \), is the number of neighbors of vertex \( v \) in \( G \), i.e. \( \deg_G(v) = |N_G(v)| \). The maximum vertex degree is \( \Delta(G) = \max_{v \in V} \deg_G(v) \). A graph \( G' = (V', E') \) is a subgraph of graph \( G = (V, E) \) if \( V' \subseteq V \) and \( E' \subseteq E \). If every edge of \( E \) with both end vertices in \( V' \) is in \( E' \), we say that \( G' \) is an induced subgraph of digraph \( G \) and we write \( G' = G[V'] \). For some undirected graph \( G = (V, E) \) its complement graph is defined by \( \overline{G} = (V, \{(u, v) \mid u, v \notin E, u, v \in V, u \neq v\}) \).

**Special Undirected Graphs** By \( P_n = (\{v_1, \ldots, v_n\}, \{(v_1, v_2), \ldots, (v_{n-1}, v_n)\}) \), \( n \geq 2 \), we denote a path on \( n \) vertices and by \( C_n = (\{v_1, \ldots, v_n\}, \{(v_1, v_2), \ldots, (v_{n-1}, v_n), (v_n, v_1)\}) \), \( n \geq 3 \), we denote a cycle on \( n \) vertices. Further by \( K_n = (\{v_1, \ldots, v_n\}, \{|v_i, v_j| \mid 1 \leq i < j \leq n\}) \), \( n \geq 1 \), we denote a complete graph on \( n \) vertices and by \( K_{n,m} = (\{v_1, \ldots, v_n, w_1, \ldots, w_m\}, \{|v_i, w_j| \mid 1 \leq i \leq n, 1 \leq j \leq m\}) \) a complete bipartite graph on \( n + m \) vertices.

**Directed Graphs** A directed graph or digraph is a pair \( D = (V, A) \), where \( V \) is a finite set of vertices and \( A \subseteq \{(u, v) \mid u, v \in V, u \neq v\} \) is a finite set of ordered pairs of distinct vertices called arcs. For a vertex \( v \in V \), the sets \( N_D^+(v) = \{u \in V \mid (u, v) \in A\} \) and \( N_D^-(v) = \{u \in V \mid (u, v) \in A\} \) are called the set of all out-neighbours and the set of all in-neighbours of \( v \). The out-degree of \( v \), \( \text{outdegree}_D(v) \) for short, is the number of out-neighbours of \( v \) and

\[ \text{outdegree}_D(v) = |N_D^+(v)| \]

\[ \text{indegree}_D(v) = |N_D^-(v)| \]

[^3]: Thus we do not consider directed graphs with loops.
the *indegree* of *v*, in-degree\(_D(v)\) for short, is the number of in-neighbours of *v* in \(D\). The *maximum out-degree* is \(\Delta^+(D) = \max_{v \in V} \text{outdegree}_D(v)\) and the *maximum in-degree* is \(\Delta^-(D) = \max_{v \in V} \text{indegree}_D(v)\). The maximum vertex degree is \(\Delta(G) = \max_{v \in V} \text{outdegree}_G(v) + \text{indegree}_G(v)\) and the *maximum semi-degree* is \(\Delta^0(G) = \max\{\Delta^-(D), \Delta^+(D)\}\).

A vertex \(v \in V\) is *out-dominating* (in-dominated) if it is adjacent to every other vertex in \(V\) and is a source (a sink, respectively). A digraph \(D' = (V', A')\) is a subdigraph of digraph \(D = (V, A)\) if \(V' \subseteq V\) and \(A' \subseteq A\). If every arc of \(D\) with both end vertices in \(V'\) is in \(A'\), we say that \(D'\) is an induced subdigraph of digraph \(D\) and we write \(D' = D[V']\). For some digraph \(D = (V, A)\) its *complement digraph* is defined by co-\(D = (V, \{(u, v) \mid (u, v) \not\in A, u, v \in V, u \neq v\})\) and its *converse digraph* is defined by con-\(D = (V, \{(u, v) \mid (v, u) \in A, u, v \in V, u \neq v\})\).

Let \(G = (V, A)\) be a digraph.
- \(G\) is *edgeless* if for all \(u, v \in V, u \neq v\), none of the two pairs \((u, v)\) and \((v, u)\) belongs to \(A\).
- \(G\) is a *tournament* if for all \(u, v \in V, u \neq v\), exactly one of the two pairs \((u, v)\) and \((v, u)\) belongs to \(A\).
- \(G\) is *semicomplete* if for all \(u, v \in V, u \neq v\), at least one of the two pairs \((u, v)\) and \((v, u)\) belongs to \(A\).
- \(G\) is *(bidirectional) complete* if for all \(u, v \in V, u \neq v\), both of the two pairs \((u, v)\) and \((v, u)\) belong to \(A\).

**Omitting the Directions** For some given digraph \(D = (V, A)\), we define its underlying undirected graph by ignoring the directions of the edges, i.e. \(\text{und}(D) = (V, \{(u, v) \mid (u, v) \in A, u, v \in V\})\).

**Orientations** There are several ways to define a digraph \(D = (V, A)\) from an undirected graph \(G = (V, E)\), see [7]. If we replace every edge \([u, v]\) of \(G\) by

- one of the arcs \((u, v)\) and \((v, u)\), we denote \(D\) as an *orientation* of \(G\). Every digraph \(D\) which can be obtained by an orientation of some undirected graph \(G\) is called an *oriented graph*.
- one or both of the arcs \((u, v)\) and \((v, u)\), we denote \(D\) as a *biorientation* of \(G\). Every digraph \(D\) which can be obtained by a biorientation of some undirected graph \(G\) is called a *bioriented graph*.
- both arcs \((u, v)\) and \((v, u)\), we denote \(D\) as a *complete biorientation* of \(G\). Since in this case \(D\) is well defined by \(G\) we also denote it by \(\overrightarrow{G}\). Every digraph \(D\) which can be obtained by a complete biorientation of some undirected graph \(G\) is called a *complete bioriented graph*.

**Special Directed Graphs** We denote by \(\overrightarrow{K}_n = ([v_1, \ldots, v_n], \{(v_i, v_j) \mid 1 \leq i \neq j \leq n\})\), \(n \geq 1\) a bioriented complete digraph on \(n\) vertices. By \(\overrightarrow{P}_n = ([v_1, \ldots, v_n], \{(v_1, v_2), \ldots, (v_{n-1}, v_n)\})\), \(n \geq 2\) we denote a directed path on \(n\) vertices and by \(\overrightarrow{C}_n = ([v_1, \ldots, v_n], \{(v_1, v_2), \ldots, (v_{n-1}, v_n), (v_n, v_1)\})\), \(n \geq 2\) we denote a directed cycle on \(n\) vertices.
An oriented forest (tree) is the orientation of a forest (tree). A digraph is an out-tree (in-tree) if it is an oriented tree in which there is exactly one vertex of indegree (outdegree) zero.

A directed acyclic digraph (DAG for short) is a digraph without any directed cycle.

3 Sequence Digraphs and Sequence Systems

3.1 From Sequences to Digraphs

Let $Q = \{q_1, \ldots, q_k\}$ be a set of $k$ sequences. Every sequence $q_i = (b_{i,1}, \ldots, b_{i,n_i})$ consists of a number $n_i$ of items, such that all $n = \sum_{i=1}^{k} n_i$ items are pairwise distinct. Further there is a function $t$ which assigns every item $b_{i,j}$ with a type $t(b_{i,j})$. The set of all types of the items in some sequence $q_i$ is denoted by $\text{types}(q_i) = \{t(b) \mid b \in q_i\}$. For a set of sequences $Q = \{q_1, \ldots, q_k\}$ we denote $\text{types}(Q) = \bigcup_{q \in Q} \text{types}(q)$. For some sequence $q_\ell = (b_{\ell,1}, \ldots, b_{\ell,n_\ell})$ we say item $b_{\ell,i}$ is on the left of item $b_{\ell,j}$ in sequence $q_\ell$ if $i < j$. Item $b_{\ell,i}$ is on the position $i$ in sequence $q_\ell$, since there are $i-1$ items on the left of $b_{\ell,i}$ in sequence $q_\ell$.

In order to insert a new item $b$ on a position $j$ in sequence $q_i$ we first move all items on positions $j' \geq j$ to position $j' + 1$ starting at the rightmost position $n_i$ and then we insert $b$ at position $j$. In order to remove an existing item $b$ at a position $j$ in sequence $q_i$ we move all items from positions $j' \geq j + 1$ to position $j' - 1$ starting at position $j + 1$.

For hardness results in Section 5.2 we consider the distribution of the items of a type $t$ onto the sequences by

$$d_Q(t) = |\{q \in Q \mid t \in \text{types}(q)\}|$$

and

$$d_Q = \max_{t \in \text{types}(Q)} d_Q(t).$$

For the number of items for type $t$ within the sequences we define

$$c_Q(t) = \sum_{q \in Q} |\{b \in q \mid t(b) = t\}|$$

and

$$c_Q = \max_{t \in \text{types}(Q)} c_Q(t).$$

Obviously it holds $d_Q \leq k$ and $1 \leq d_Q \leq c_Q \leq n$.

**Definition 1** (Sequence Digraph) The sequence digraph $g(Q) = (V, A)$ for a set $Q = \{q_1, \ldots, q_k\}$ has a vertex for every type, i.e. $V = \text{types}(Q)$ and an arc $(u, v) \in A$ if and only if there is some sequence $q_i$ in $Q$ where an item of type $u$ is on the left of an item of type $v$. More formally, there is an arc $(u, v) \in A$ if and only if there is some sequence $q_i$ in $Q$, such that there are two items $b_{i,j}$ and $b_{i,j'}$ such that

1. $1 \leq j < j' \leq n_i$,
2. $t(b_{i,j}) = u$,
3. $t(b_{i,j'}) = v$, and
4. $u \neq v$.  

© Springer
Sequence digraphs have successfully been applied in order to model the stacking process of bins from conveyor belts onto pallets with respect to customer orders, which is an important task in palletizing systems used in centralized distribution centers [8]. In this paper we show graph theoretic properties of sequence digraphs and their relation to special digraph classes.

In our examples we will use type identifications instead of item identifications to represent a sequence $q_i \in Q$. For (not necessarily distinct) $r$ types $t_1, \ldots, t_r$ let $[t_1, \ldots, t_r]$ denote some sequence $q_i = (b_{i,1}, \ldots, b_{i,r})$ of $r$ pairwise distinct items, such that $t(b_{i,j}) = t_j$ for $j = 1, \ldots, r$. We use this notation for sets of sequences as well.

Example 1 (Sequence Digraph) Figure 1 shows the sequence digraph $g(Q)$ for $Q = \{q_1, q_2, q_3\}$ with sequences $q_1 = [a, a, d, e, d]$, $q_2 = [c, b, b, d]$, and $q_3 = [c, c, d, e, d]$.

In Example 1 all sequences of $Q$ contain consecutive items of the same type. This is obviously not necessary to obtain digraph $g(Q)$. Let $C(q_i)$ be the subsequence of $q_i$ which is obtained from $q_i$ by removing all but one of consecutive items of the same type for each type and $C(Q) = \{C(q_1), \ldots, C(q_k)\}$.

Observation 1 Let $Q = \{q_1, \ldots, q_k\}$ be a set of $k$ sequences, then $g(Q) = g(C(Q))$.

Next we give methods in order to compute the sequence digraph and also its complement digraph. Therefore we define the position of the first item in some sequence $q_i \in Q$ of some type $t \in \text{types}(Q)$ by $\text{first}(q_i, t)$ and the position of the last item of type $t$ in sequence $q_i$ by $\text{last}(q_i, t)$. For technical reasons, if there is no item for type $t$ contained in sequence $q_i$, then we define $\text{first}(q_i, t) = n_i + 1$, and $\text{last}(q_i, t) = 0$.

Lemma 1 Let $Q = \{q_1\}$ be some set of one sequence, $g(Q) = (V, A)$ the defined sequence digraph, $\text{co-(}g(Q)) = (V, A^c)$ its complement digraph, and $u \neq v$ two vertices of $V$.

1. There is an arc $(u, v) \in A$, if and only if $\text{first}(q_1, u) < \text{last}(q_1, v)$.
2. There is an arc $(u, v) \in A^c$, if and only if $\text{last}(q_1, v) < \text{first}(q_1, u)$.

Fig. 1 Sequence digraph $g(Q)$ of Example 1
3. If \((u, v) \in A^c\), then \((v, u) \in A\).
4. There is an arc \((u, v) \in A\) and an arc \((v, u) \in A^c\), if and only if \(\text{last}(q_1, u) < \text{first}(q_1, v)\).

**Proof**

1. By the definition of the arcs of sequence digraphs.
2. It holds \((u, v) \in A^c\) if and only if \((u, v) \notin A\). Further by (1) and \(u \neq v\) it holds \((u, v) \notin A\) if and only if \(\text{first}(q_1, u) > \text{last}(q_1, v)\).
3. If \((u, v) \in A^c\), then our result (2) implies \(\text{last}(q_1, v) < \text{first}(q_1, u)\) which implies that \((v, u) \in A\).
4. Follows by (2) and (3).

**Lemma 2** Let \(Q = \{q_1, \ldots, q_k\}\) be some set of \(k\) sequences, \(g(Q) = (V, A)\) the defined sequence digraph, \(\text{co-}g(Q) = (V, A^c)\) its complement digraph, and \(u \neq v\) two vertices of \(V\).

1. There is an arc \((u, v) \in A\), if and only if there is some \(q_i \in Q\) such that \(\text{first}(q_i, u) < \text{last}(q_i, v)\).
2. There is an arc \((u, v) \in A^c\), if and only if for every \(q_i \in Q\) it holds \(\text{last}(q_i, v) < \text{first}(q_i, u)\).

**Proof**

1. By the definition of the arcs of sequence digraphs.
2. It holds \((u, v) \in A^c\) if and only if \((u, v) \notin A\). Further by (1) and \(u \neq v\) we have \((u, v) \notin A\) if and only if for every \(q_i \in Q\) it holds \(\text{first}(q_i, u) > \text{last}(q_i, v)\). 

Properties (3) and (4) of Lemma 1 can not be shown for \(k > 1\) sequences, since an arc \((u, v) \in A^c\) can be obtained by two types \(u\) and \(v\) from two different sequences \(q_i\) and \(q_j\) such that \(u \in \text{types}(q_i) - \text{types}(q_j)\) and \(v \in \text{types}(q_j) - \text{types}(q_i)\). In such situations it is not necessary to have \((v, u) \notin A\).

By Lemma 2 (1) only the first and the last item of each type in every \(q_i \in Q\) are important for the arcs in the corresponding digraph. Let \(M(q_i)\) be the subsequence of \(q_i\) which is obtained from \(q_i\) by removing all except the first and last item for each type and \(M(Q) = \{M(q_1), \ldots, M(q_k)\}\).

**Observation 2** Let \(Q = \{q_1, \ldots, q_k\}\) be a set of \(k\) sequences, then \(g(Q) = g(M(Q))\).

Using Lemma 2 and Observation 2 the running time to obtain the sequence digraph can be bounded as follows.

**Proposition 1** Given a set \(Q\) of sequences, the sequence digraph \(g(Q) = (V, A)\) can be computed in time \(O(n + k \cdot |\text{types}(Q)|^2) = O(n + k \cdot |V|^2)\).
First, we initialize the values \( \text{first}(q_i, t) := |q_i| \) and \( \text{last}(q_i, t) := 0 \) for each sequence \( q_i \) and type \( t \). Then, we compute the values \( \text{first}(q_i, t) \) and \( \text{last}(q_i, t) \) by scanning each sequence \( q_i \) from left to right and at position \( j \) where \( t(b_{i,j}) = t \) set the values \( \text{last}(q_i, t) := j \) and \( \text{first}(q_i, t) := j \). The last assignment is made only if \( \text{first}(q_i, t) = |q_i| \) has its initial value. Finally, \( g(Q) \) can be computed in \( O(k \cdot |V|^2) \) time by considering each pair \( t, t' \) of types for every sequence \( q_i \) and add the arc \((t, t')\) to \( A \) if \( \text{first}(q_i, t) < \text{last}(q_i, t') \).

### 3.2 From Digraphs to Sequences

**Definition 2** (Sequence System) Let \( G = (V, A) \) be some digraph and \( A = \{a_1, \ldots, a_\ell\} \) its arc set. The sequence system \( q(G) = \{q_1, \ldots, q_\ell\} \) for \( G \) is defined as follows.

1. There are \( 2\ell \) items \( b_{1,1}, b_{1,2}, \ldots, b_{\ell,1}, b_{\ell,2} \).
2. Sequence \( q_i = (b_{i,1}, b_{i,2}) \) for \( 1 \leq i \leq \ell \).
3. The type of item \( b_{i,1} \) is the first vertex of arc \( a_i \) and the type of item \( b_{i,2} \) is the second vertex of arc \( a_i \) for \( 1 \leq i \leq \ell \). Thus \( \text{types}(q(G)) = V \).

**Example 2** (Sequence System) For the digraph \( G \) of Fig. 2 the corresponding sequence system is given by \( q(G) = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7\} \), where \( q_1 = [a, b], q_2 = [b, c], q_3 = [c, d], q_4 = [d, e], q_5 = [e, a], q_6 = [e, f], q_7 = [f, a] \). The sequence digraph of \( q(G) \) is \( G \).

The definitions of the sequence system \( q(G) \) and the sequence digraph \( g(Q) \), defined in Section 3.1, imply the following results.

**Observation 3** For every digraph \( G \) it holds \( G = g(q(G)) \).

The related relation \( Q = q(g(Q)) \) is not true in general (e.g. not for \( Q \) from Example 1). But if the reduction to different consecutive items \( C(Q) \) contains exactly two items of different types the following equivalence holds true.

![Fig. 2 Digraph G of Example 2](image)

© Springer
Lemma 3 For every set $Q$ of sequences it holds $C(Q) = q(g(Q))$ if and only if each sequence $q_i \in C(Q)$ contains exactly two items of different types.

Since $q(g(Q)) = q(g(C(Q)))$ holds for every set of sequences $Q$, in Lemma 3 we also can replace condition $C(Q) = q(g(Q))$ by condition $C(Q) = q(g(C(Q)))$.

Lemma 4 For every digraph $G = (V, A)$ with $\text{und}(G) = (V, E)$ there is a set $Q$ of at most $|E|$ sequences such that $G = g(Q)$.

Proof For every digraph $G = (V, A)$ by Observation 3 the sequence system $Q = q(G)$ leads to a set of at most $|A|$ sequences such that $G = g(Q)$. If we have two arcs $(u, v)$ and $(v, u)$ between two vertices $u$ and $v$ these can be represented by one sequence $[u, v, u]$. 

There are digraphs which even can be defined by one sequence (see Example 3 (1) and Theorem 1) and there are digraphs for which $|E|$ sequences are really necessary (see Lemmas 10 and 9).

For digraphs of bounded vertex degree the sequence system $Q = q(G)$ leads to sets whose distribution and number of items of each type can be bounded as follows.

Lemma 5 For every digraph $G = (V, A)$ where $\max(\Delta^- (G), \Delta^+ (G)) \leq d$ there is a set $Q$ with $d_Q \leq 2d$ and $c_Q \leq 2d$ such that $G = g(Q)$.

In case of complete bioriented digraphs we can improve the latter bounds as follows.

Lemma 6 For every complete bioriented digraph $G = (V, A)$ where $\max(\Delta^- (G), \Delta^+ (G)) \leq d$ there is a set $Q$ with $d_Q \leq d$ and $c_Q \leq 2d$ (for $d \geq 2$ even $c_Q \leq 2d - 1$) such that $G = g(Q)$.

Proof By Lemma 5 we obtain $d_Q \leq 2d$ and $c_Q \leq 2d$. Since we consider complete bioriented digraphs for every two vertices $u$ and $v$ we now have two sequences $[u, v]$ and $[v, u]$, which can be combined to only one sequence $[u, v, u]$. This leads to $d_Q \leq d$ and $c_Q \leq 2d$. For $d \geq 2$ we can show the stated improvement as follows. If there is one type $t \in \text{types}(Q)$ such that there are exactly $2d$ items of type $t$, then there are less than $d$ types $t' \in \text{types}(Q)$ such that there are exactly $2d - 1$ items of type $t'$. Thus there is one sequence $[t, t'', t]$ such that there are at most $2d - 2$ items of type $t''$. If we substitute $[t, t'', t]$ by $[t'', t, t']$ we reduced the number of items of type $t$ to $2d - 1$. If we repeat this substitution until there is no type $t \in \text{types}(Q)$ such that there are exactly $2d$ items of type $t$, we achieve $c_Q \leq 2d - 1$. 

\[\square\]
4 Properties of Sequence Digraphs

4.1 Graph Classes and their Relations

In order to represent some digraph $G = (V, A)$ as a sequence digraph we need exactly $|V|$ types. By Lemma 4 every digraph is a sequence digraph using a suitable set $Q$ of sequences. Thus we want to consider digraphs which can be defined by a given upper bound for the number of sequences. Furthermore the first and the last item of each type within a sequence is the most important one by Observation 2, thus we want to analyze the digraphs which can be defined by a given upper bound for the number of items of each type. We define $S_{k,\ell}$ to be the set of all sequence digraphs defined by sets $Q$ on at most $k$ sequences that together contain at most $\ell$ items of each type in $\text{types}(Q)$. By the definition we know for every two integers $k \geq 1$ and $\ell \geq 1$ the following inclusions between these graph classes.

\[
S_{k,\ell} \subseteq S_{k+1,\ell} \quad (1) \\
S_{k,\ell} \subseteq S_{k,\ell+1} \quad (2)
\]

By Lemma 4 and Observation 2 we obtain the following bounds.

Corollary 1 Let $Q$ be a set on $k$ sequences and $g(Q) = (V, A) \in S_{k,\ell}$ the defined graph with $\text{und}(g(Q)) = (V, E)$. Then, $g(Q)$ in $S_{k,\ell'}$ for all $\ell' \geq 2k$ and $g(Q)$ in $S_{k',\ell}$ for all $k' \geq |E|$.

Lemma 7 Let $\ell \geq 1$ and $G \in S_{1,\ell}$ be defined by $Q = \{q_1\}$, then digraph $g(Q)$ is semicomplete and thus graph $\text{und}(g(Q))$ is the complete graph on $|\text{types}(Q)|$ vertices.

Proof Let $Q = \{q_1\}$ be some set of one sequence which defines the sequence digraph $g(Q)$. For every vertex $v$ in $g(Q)$ there is exactly one corresponding type $t_v \in \text{types}(q_1)$. Since $G \in S_{1,\ell}$ for every two vertices $v$ and $w$ in $g(Q)$ there are two items $b_{1,i}$ and $b_{1,j}$ of type $t_v$ and $t_w$, respectively, which define the arc $(v, w)$ or $(w, v)$.

Corollary 2 Let $G = (V, A)$ be a digraph, such that graph $\text{und}(G)$ has $k$ connected components. Then for every $k' < k$ and $\ell \geq 1$ it holds $G \not\in S_{k',\ell}$.

Next we consider the relations of the defined classes for $k = 1$ sequence. Since $S_{1,1}$ contains only digraphs with exactly one arc between every pair of vertices (cf. Theorem 1 for a more precise characterization) and $S_{1,\ell}$ for $\ell \geq 2$ contains all bidirectional complete graphs we know that $S_{1,1} \subsetneq S_{1,\ell}$ for every $\ell \geq 2$. Further by (2) and by Observation 2 it follows that all classes $S_{1,\ell}$ for $\ell \geq 2$ are equal.
Lemma 8  For every two integers $\ell, \ell' \geq 2$ it holds $S_{1, \ell} = S_{1, \ell'}$.

Corollary 3  For $\ell \geq 2$ the following inclusions hold.

$$S_{1,1} \subseteq S_{1,2} = \ldots = S_{1,\ell}$$

The equalities of Lemma 8 can not be generalized for $k > 1$ sequences and $\ell \leq k$ since removing items from different sequences can change the sequence digraph. In order to give examples for digraphs which do not belong to some of the classes $S_{k,\ell}$ we next show some useful results.

For a set of digraphs $\mathcal{F}$ we denote by $\mathcal{F}$-free digraphs the set of all digraphs $G$ such that no induced subdigraph of $G$ is isomorphic to a member of $\mathcal{F}$. If $\mathcal{F}$ consists of only one digraph $F$, we write $F$-free instead of $\{F\}$-free. For undirected graphs we use this notation as well.

Lemma 9  Let $G = (V, E)$ be a triangle free graph, i.e. a $C_3$-free graph, with $|E| \geq 2$, such that $\Delta(G) = \ell$ and $G' = (V, A)$ be an orientation of $G$. Then for $k = |E|$ it holds $G' \in S_{k,\ell}$ but for $k' < k$ or $\ell' < \ell$ it holds $G' \notin S_{k',\ell'}$.

Proof  Let $G$ and $G'$ be as in the statement of the lemma. Further let $Q$ be a set of sequences such that $G' = g(Q)$. If some sequence $q_i \in Q$ defines more than one arc of $G'$, then $q_i$ has to contain at least three items of pairwise distinct types. This would induce a $C_3$ in $G$, which is not possible by our assumption. Thus we have to represent every arc $(u, v)$ of $G'$ by one sequence $[u, v]$. Thus $G'$ can not be defined by less than $k = |E|$ sequences or less than $\ell = \Delta(G)$ items for every type. \qed

Since for every $k \geq 2$ and every $\ell = 2, \ldots, k$ there is a tree $T$ on $k$ edges and $\Delta(T) = \ell$ we know by Lemma 9 that for $k \geq 2$ and $\ell = 2, \ldots, k$ we have $S_{k,\ell-1} \subset S_{k,\ell}$. Further by Observation 2 we know that for $k \geq 2$ and $\ell \geq 2k$ it holds $S_{k,\ell} = S_{k,\ell+1}$.

Corollary 4  For $k \geq 2$ the following inclusions hold.

$$S_{k,1} \subseteq S_{k,2} \subseteq \ldots \subseteq S_{k,k} \subseteq S_{k,k+1} \subseteq \ldots \subseteq S_{k,2k} = S_{k,2k+1} = \ldots$$

Proposition 2  Let $G \in S_{k,\ell}$, then for every induced subdigraph $H$ of $G$ it holds $H \in S_{k,\ell}$.

Proof  A set of sequences $Q'$ for an induced subdigraph $H$ can be obtained from a set $Q$ of the original digraph $G$ by restricting $Q'$ to the items which are destined for types corresponding to vertices of $H$. \qed

Graph classes which are closed under taking induced subgraphs are called hereditary. Hereditary graph classes are exactly those classes which can be defined by a set of forbidden induced subgraphs. On the other hand, the classes $S_{k,\ell}$ are not closed under taking arbitrary subgraphs by the following example.
Example 3 1. For every $n \geq 1$ and $\ell \geq 2$ it holds $\overrightarrow{K_n} \in S_{1,\ell}$, which can be verified by set $Q = \{q_1\}$, where

$$q_1 = [v_1, v_2, \ldots, v_n, v_1, v_2, \ldots, v_n].$$

2. Every orientation $T'_{n,d}$ of a tree $T_{n,d}$ on $n$ vertices and $\Delta(T) = d \leq n - 1$ can be obtained as a subdigraph of $\overrightarrow{K_n}$. Since by Lemma 9 for $n \geq 3$ it holds $T'_{n,d} \in S_{n-1,d}$ but $T'_{n,d} \notin S_{n-2,d}$ and $T'_{n,d} \notin S_{n-1,d-1}$ the classes $S_{k,\ell}$ for each $k \geq 1$ and $\ell \geq 2$ are not closed under taking arbitrary subgraphs.

3. By Theorem 2 digraph $TT_k$ which consists of the disjoint union of $k$ transitive tournaments is in $S_{k,1} = S_{k-1,1}$.

4. If we remove exactly one edge from $TT_k$ we obtain a digraph $TT'_k$ such that $TT'_k \notin S_{k,1}$. Thus the classes $S_{k,1}$ for every $k \geq 1$ are not closed under taking arbitrary subgraphs.

For orientations of paths and cycles Lemma 9 leads to the following bounds.

Lemma 10 1. Let $P'_n$ be an orientation of a path $P_n$ on $n \geq 2$ vertices, e.g. $P'_n = \overrightarrow{P_n}$ for $n \geq 2$. Then for every $\ell \geq 2$ it holds $P'_n \in S_{n-1,\ell}$. But for every $\ell \geq 1$ it holds $P'_n \notin S_{n-2,\ell}$.

2. Let $C'_n$ be an orientation of a cycle $C_n$ on $n \geq 4$ vertices, e.g. $C'_n = \overrightarrow{C_n}$ for $n \geq 4$. Then for every $\ell \geq 2$ it holds $C'_n \in S_{n,\ell}$. But for every $\ell \geq 1$ it holds $C'_n \notin S_{n-1,\ell}$.

By Lemma 10 and Proposition 2 we obtain the following results.

Proposition 3 1. Let $\ell \geq 1$, $k \geq 1$, and $G \in S_{k,\ell}$, then $G$ has no induced path $\overrightarrow{P_k'}$ for $k' \geq k + 2$.

2. Let $\ell \geq 1$, $k \geq 3$, and $G \in S_{k,\ell}$, then $G$ has no induced cycle $\overrightarrow{C_k'}$ for $k' \geq k + 1$.

4.2 Characterizations of Sequence Digraphs for $k = 1$ or $\ell = 1$

By Proposition 2 all the classes $S_{k,\ell}$ are hereditary and thus can be characterized by a set of forbidden induced subgraphs. In this section we even show a finite set of forbidden induced subgraphs for all classes $S_{k,\ell}$ where $\ell = 1$ and for all classes where $k = 1$. These characterizations lead to polynomial time recognition algorithms for the corresponding graph classes. Furthermore we give characterizations in terms of special tournaments and conditions for the complement digraph.

4.2.1 Sequence Digraphs for $k = 1$ and $\ell = 1$

Regarding the algorithmic use of special digraphs it is often helpful to know whether they are transitive. A digraph $G = (V, A)$ is called transitive if for every pair $(u, v) \in A$ and $(v, w) \in A$ of arcs with $u \neq w$ the arc $(u, w)$ also belongs to $A$. 

© Springer
Lemma 11 Every digraph in $S_{1,1}$ is transitive.

Proof Let $Q = \{q_1\}$ be some set of one sequence. If for every type there is only one item in $q_1$, then two arcs $(u, v)$ and $(v, w)$ can only be defined in $g(Q)$ if $q_1$ contains some item $b_{1,i}$ of type $u$, some item $b_{1,i'}$ of type $v$, and some item $b_{1,i''}$ of type $w$ such that $i < i'$ and $i' < i''$. By the definition of the sequence digraph $g(Q)$ also has the arc $(u, w)$.

For some digraph $G$ and some integer $d$ let $dG$ be the disjoint union of $d$ copies of $G$.

Theorem 1 For every digraph $G$ the following statements are equivalent.

1. $G \in S_{1,1}$
2. $G$ is a transitive tournament.
3. $G$ is an acyclic tournament.
4. $G$ is a $\overrightarrow{C}_3$-free tournament.
5. $G$ is a tournament with exactly one Hamiltonian path.
6. $G$ is a tournament and every vertex in $G$ has a different outdegree, i.e. $\{\text{outdegree}(v) \mid v \in V\} = \{0, \ldots, |V|-1\}$.
7. $G$ is $\{\overrightarrow{2K_1}, \overrightarrow{K_2}, \overrightarrow{C_3}\}$-free.
8. $G$ can be constructed from the one-vertex graph $K_1$ by repeatedly adding an out-dominating vertex or by repeatedly adding an in-dominated vertex.

Proof The equivalence of (2)–(6) is known from [9, Chapter 9].

(1) $\Rightarrow$ (2) By Lemma 11 digraph in $S_{1,1}$ is transitive and by definition of $S_{1,1}$ digraph $G$ is a tournament.

(3) $\Rightarrow$ (1) Every acyclic digraph $G$ has a source, i.e. a vertex $v_1$ of indegree 0, see [7]. Since $G$ is a tournament there is an arc $(v_1, v)$ for every vertex $v$ of $G$, i.e. $v_1$ is an out-dominating vertex. By removing $v_1$ from $G$, we obtain a transitive tournament $G^1$ which leads to an out-dominating vertex $v_2$. By removing $v_2$ from $G^1$, we obtain a transitive tournament $G^2$ which leads to an out-dominating vertex $v_3$ and so on. The sequence $[v_1, v_2, \ldots, v_n]$ shows that $G \in S_{1,1}$.

(4) $\iff$ (7) Obviously.

(1) $\iff$ (8) holds by definition.

By the method is given in part (3) $\Rightarrow$ (1) of the proof of Theorem 1 we obtain the following result.

Proposition 4 Let $G = (V, A) \in S_{1,1}$, then a set $Q$ of one sequence $q$, such that $G = g(Q)$ can be found in time $O(|V| + |A|)$. 
4.2.2 Sequence Digraphs for $\ell = 1$

The sequence digraph $g(Q) = (V, A)$ for a set $Q = \{q_1, \ldots, q_k\}$ can be obtained by the union of $g(q_i) = (V_i, A_i)$, $1 \leq i \leq k$ by $V = \bigcup_{i=1}^{k} V_i$ and $A = \bigcup_{i=1}^{k} A_i$. Since for digraphs in $S_{k,1}$ the vertex sets $V_i = \text{types}(q_i)$ are disjoint, we can follow the next lemma.

**Lemma 12** For every integer $k$ every digraph in $S_{k,1}$ is the disjoint union of $k$ digraphs in $S_{1,1}$.

This lemma allows us to generalize Lemma 11 to $k$ sequences.

**Corollary 5** For every integer $k$ every digraph in $S_{k,1}$ is transitive.

On the other hand none of the sets $S_{k,1}$ contains all transitive digraphs. Therefore let $T'$ be an orientation of a tree $T$ with $k+1$ edges, such that for every vertex $v$ in $T'$ either the indegree or outdegree is 0. Such an orientation is possible for every tree $T$ and leads to a transitive graph $T'$. By Lemma 9 we know that $T' \notin S_{k,1}$.

Theorem 1 can be generalized to $k \geq 1$ sequences as follows.

**Theorem 2** For every digraph $G$ and every integer $k \geq 1$ the following statements are equivalent.

1. $G \in S_{k,1}$.
2. $G$ is the disjoint union of $k$ digraphs from $S_{1,1}$.
3. $G$ is the disjoint union of $k$ transitive tournaments.
4. $G$ is the disjoint union of $k$ acyclic tournaments.
5. $G$ is the disjoint union of $k \overrightarrow{C}_3$-free tournaments.
6. $G$ is $\{((k+1)\overrightarrow{K}_1, \overrightarrow{K}_2, \overrightarrow{C}_3)\}$-free.

**Proposition 5** Let $G = (V, A) \in S_{k,1}$, then a set $Q$ on $k$ sequences, such that $G = g(Q)$ can be found in time $O(|V| + |A|)$.

**Proof** If $G \in S_{k,1}$ then $G$ is the disjoint union of $k$ digraphs from $S_{1,1}$. For every of them we can define a sequence $q_i$ by Proposition 4. \qed

4.2.3 Sequence Digraphs for $k = 1$

The next examples show that for $\ell \geq 2$ items for each type even one sequence can define digraphs which are not transitive.
Example 4 1. The digraph $D_0$ in Table 2 is not transitive, since it has among others the arcs $(b, c)$ and $(c, a)$ but not the arc $(b, a)$. Further $D_0$ belongs to the set $S_{1,2}$, since it can be defined by set $Q = \{q_1\}$ of one sequence $q_1 = [c, a, b, c]$.
2. The digraph $D_6$ in Table 2 is not transitive, since it has among others the arcs $(c, b)$ and $(b, a)$ but not the arc $(c, a)$. Further $D_6$ belongs to the set $S_{1,2}$, since it can be defined by set $Q = \{q_1\}$ of one sequence $q_1 = [a, b, a, c, b, d, c, d]$.

In order to show a characterization for the class $S_{1,2}$ we next prove some lemmas.

Lemma 13 Let $\ell \geq 1$ and $G \in S_{1,\ell}$, then the complement digraph $\overline{G}$ is transitive.

Proof Let $Q = \{q_1\}$ be some sequence which defines the sequence digraph $g(Q)$ and $(u, v)$ and $(v, w)$ with $u \neq w$ be two arcs of co-$(g(Q))$. By Lemma 1 (2) we conclude that $\text{last}(q_1, v) < \text{first}(q_1, u)$ and $\text{last}(q_1, w) < \text{first}(q_1, v)$ thus we have $\text{last}(q_1, w) < \text{first}(q_1, v) \leq \text{last}(q_1, v) < \text{first}(q_1, u)$ and $\text{last}(q_1, w) < \text{first}(q_1, u)$ implies by Lemma 1 (2) that arc $(u, w)$ is in digraph co-$(g(Q))$. $\square$

Lemma 14 Let $\ell \geq 1$ and $G \in S_{1,\ell}$, then the complement digraph $\overline{G}$ is $\overrightarrow{2P_2}$-free.

Proof If we assume that co-$G$ contains $\overrightarrow{2P_2} = \{(a, b, c, d), ((c, a), (d, b))\}$, see Table 2, as an induced subdigraph, then $G$ has (among others) the arcs $(c, b)$ and $(d, a)$. By applying Lemma 1 (1) and (2) we obtain $\text{last}(q_1, a) < \text{first}(q_1, c) < \text{last}(q_1, b) < \text{first}(q_1, d) < \text{last}(q_1, a)$, which leads to a contradiction. $\square$

Lemma 15 Let $G$ be a semicomplete $(\overrightarrow{C_3}, D_4)$-free digraph on $n$ vertices, then $G$ has a vertex $v$ such that outdegree$(v) = n - 1$ and a vertex $v'$ such that indegree$(v') = n - 1$. ($D_4$ is shown in Table 2.)

Proof We show the result by induction on the number of vertices $n$ in digraph $G$. For digraphs on $n \leq 3$ vertices the statement is true.$^4$

Let $n \geq 3$ and $G_{n+1} = (V_{n+1}, A_{n+1})$ be a semicomplete $(\overrightarrow{C_3}, D_4)$-free digraph on $n + 1$ vertices. Further let $w \in V_{n+1}$ and $G_n = G_{n+1}[V_{n+1} - \{w\}]$. Then $G_n = (V_n, A_n)$ is a semicomplete $(\overrightarrow{C_3}, D_4)$-free digraph on $n$ vertices. By the induction hypothesis there is some $v \in V_n$ such that outdegree$(v) = n - 1$ in $G_n$.

If $(v, w) \in A_{n+1}$ then outdegree$(v) = n$ in $G_{n+1}$ and the proof is done. If $(v, w) \notin A_{n+1}$ then the semicompleteness of $G_{n+1}$ implies that $(w, v) \in A_{n+1}$. Then for every $(v, u) \in E_n$ it holds $(w, u) \in E_{n+1}$. Otherwise the semicompleteness of $G_{n+1}$ implies that $(u, w) \in E_{n+1}$ and thus $G_{n+1}[[u, v, w]]$ leads to a $\overrightarrow{C_3}$ or to a $D_4$. Thus in $G_{n+1}$ it holds outdegree$(w) = \text{outdegree}(v) + 1 = n - 1 + 1 = n$.

$^4$A list of all digraphs on at most four vertices can be found in Appendix 2 of [10].
The proof for the existence of a vertex $v'$ such that indegree$(v') = n - 1$ is similar.

Lemma 16 Let $G$ be a semicomplete $\overrightarrow{C_3}, D_4$-free digraph, then its complement digraph $\text{co-}G$ is transitive. ($D_4$ is shown in Table 2.)

Proof Let $(u, v), (v, w) \in A^c$ be two arcs of $\text{co-}G = (V, A^c)$. Since $G = (V, A)$ is semicomplete we know that $(v, u), (w, v) \notin A^c$. If $G$ is a $\overrightarrow{C_3}, D_4$-free digraph, then $\text{co-}G$ is a $\overrightarrow{C_3}, P_3$-free digraph. Thus $u$ and $w$ are connected either only by $(u, w) \in A^c$ or by $(u, w) \in A^c$ and $(w, u) \in A^c$, which implies that $\text{co-}G$ is transitive.

A spanning subdigraph is a subdigraph obtained by the deletion of arcs.

Lemma 17 Every semicomplete $\overrightarrow{C_3}, D_4$-free digraph has a spanning transitive tournament subdigraph. ($D_4$ is shown in Table 2.)

Proof By Lemma 15 every semicomplete $\overrightarrow{C_3}, D_4$-free digraph $G$ on $n$ vertices has a vertex $v_1$ such that outdegree$(v_1) = n - 1$. By removing $v_1$ from $G$, we obtain a semicomplete $\overrightarrow{C_3}, D_4$-free digraph $G^1$ on $n - 1$ vertices, which leads to a vertex $v_2$ such that outdegree$(v_2) = n - 2$. By removing $v_2$ from $G^1$, we obtain a semicomplete $\overrightarrow{C_3}, D_4$-free digraph $G^2$ on $n - 2$ vertices, which leads to a vertex $v_3$ such that outdegree$(v_3) = n - 3$ and so on. The sequence $q = [v_1, v_2, \ldots, v_n]$ defines a digraph $g((q)) \in S_1,1$ which is a spanning transitive tournament subdigraph of $G$.

These results allow us to show the following characterizations for the class $S_{1,2}$. Since we use several forbidden induced subdigraphs the semicompleteness is expressed by excluding $2\overrightarrow{K_1}$.

Theorem 3 For every digraph $G$ the following statements are equivalent.

1. $G \in S_{1,2}$
2. $G \in S_{1,\ell}$ for some $\ell \geq 2$
3. $\text{co-}G$ is transitive, $\text{co-}G$ is $2\overrightarrow{P_2}$-free, and $G$ has a spanning transitive tournament subdigraph.
4. $G$ is $\{\text{co-}(2\overrightarrow{P_2}), 2\overrightarrow{K_1}, \overrightarrow{C_3}, D_4\}$-free. ($D_4$ is shown in Table 2.)

Proof (1) $\Rightarrow$ (2) By definition.
(2) $\Rightarrow$ (1) By Lemma 8.
(3) $\Rightarrow$ (4) If $G$ has a spanning transitive tournament subdigraph then it is semicomplete and thus $2\overrightarrow{K_1}$-free. If $\text{co-}G$ is transitive it has no induced $\overrightarrow{C_3}$ and no $\overrightarrow{P_3}$. Thus $G$ has no induced $\overrightarrow{C_3}$ and $D_4$. (4) $\Rightarrow$ (3) Follows by Lemmas 16 and 17.
(1) ⇒ (3) Digraph co-$G$ is transitive by Lemma 13 and $2P_2$-free by Lemma 14. Further if $G$ is defined by sequence $q$ then the subsequence $F(q)$ which is obtained from $q$ by removing all except the first item for each type leads to a subdigraph $(V, A')$ which is a transitive tournament.

(3) ⇒ (2) Let $G' = (V, A')$ be a subdigraph of $G = (V, A)$ which is a transitive tournament. By Theorem 1 we know that $G' \in S_{1,1}$ and thus there is some sequence $q' = [v_1, \ldots, v_n]$ such that $g([q']) = G'$.

If for every arc $(vi, vj)$ we can assume that $G$ tournament. By Theorem 1 we know that $G' \in S_{1,1}$ and thus there is some sequence $q' = [v_1, \ldots, v_n]$ such that $g([q']) = G'$. If $A' = A$ we know that $G \in S_{1,1} \subseteq S_{1,\ell}$ for every $\ell \geq 2$. So we can assume that $A' \subseteq A$. Obviously for every arc $(vi, vj) \in A - A'$ there are two positions $j < i$ in $q' = [v_1, \ldots, v_j, \ldots, v_i, \ldots, v_n]$.

In order to define a subdigraph of $G$ which contains all arcs of $G'$ and arc $(vi, vj)$ we can insert (cf. Section 3.1 for the definition of inserting an item) an additional item for type $vi$ on position $k \leq j$, or an additional item for type $vj$ on position $k > i$, or first an additional item for type $vj$ and then an additional item for type $vi$ on a position $k, j < k \leq i$, into $q'$ without creating an arc which is not in $A$. This is possible if and only if there is some position $k, j \leq k \leq i$, in $q' = [v_1, \ldots, v_j, \ldots, v_m', \ldots, v_k, \ldots, v_m'', \ldots, v_i, \ldots, v_n]$ such that for every $m', j < m' \leq k$, it holds $(v_m', v_j) \in A$ and for every $m''$, $k \leq m'' < i$, it holds $(v_i, v_{m''}) \in A$.

If it is possible to insert all arcs of $A - A'$ by adding a set of additional items into sequence $q'$ resulting in a sequence $q$ such that $G = g(q)$, then we have $G \in S_{1,\ell}$ for some $\ell \geq 2$. Next we show a condition using the new items of every single arc of $A - A'$ independently from each other.

**Claim 1** If for every arc $(vi, vj) \in A - A'$ there is a position $k, j < k \leq i$ such that first inserting an additional item for type $vj$ and then an additional item for type $vi$ at position $k$ into $q'$ defines a subdigraph of $G$ which contains all arcs of $G'$ and arc $(vi, vj)$, then $G \in S_{1,\ell}$ for some $\ell \geq 2$.

**Proof** (of Claim 1) Every single arc $(vi, vj) \in A - A'$ can be inserted by first inserting first an additional item for type $vj$ and then an additional item for type $vi$ at some position $k, j < k \leq i$. Next we show that the new inserted items for two missing arcs $(vi, vj) \in A - A'$ at position $k$ and $(v'_i, v'_j) \in A - A'$ at position $k'$ do not create an arc which is not in $A$. This is done by a case distinction w.r.t. the 34 possible positions of $i, i', j, j', k, k'$ shown in Table 3.

1. If $j' < k' < i' < j < k < i$, then $(v'_i, vi) \in A, (v'_i, v_i) \in A, (v'_j, v_j) \in A$, and $(v'_j, v_j) \in A$.
2. If $j' < k' < i' = j < k < i$, then $(v'_i, vi) \in A, (v'_i, v_j) \in A, and (v'_j, v_j) \in A$.
3. If $j' < k' < j < i' < k < i$, then $(v'_j, v_i) \in A, (v'_i, v_i) \in A, (v'_j, v_j) \in A$, and $(v'_j, v_j) \in A$. 


### Table 3  Cases within the proof of Claim 1

|   | $v_j$  | $v_k$  | $v_i$  |
|---|--------|--------|--------|
| 1 | $v'_j$  | $v'_k$  | $v'_i$  |
| 2 | $v'_j$  | $v'_k$  | $v'_i$  |
| 3 | $v'_j$  | $v'_k$  | $v'_i$  |
| 4 | $v'_j$  | $v'_k$  | $v'_i$  |
| 5 | $v'_j$  | $v'_k$  | $v'_i$  |
| 6 | $v'_j$  | $v'_k$  | $v'_i$  |
| 7 | $v'_j$  | $v'_k$  | $v'_i$  |
| 8 | $v'_j$  | $v'_k$  | $v'_i$  |
| 9 | $v'_j$  | $v'_k$  | $v'_i$  |
| 10 | $v'_j$  | $v'_k$  | $v'_i$  |
| 11 | $v'_j$  | $v'_k$  | $v'_i$  |
| 12 | $v'_j$  | $v'_k$  | $v'_i$  |
| 13 | $v'_j$  | $v'_k$  | $v'_i$  |
| 14 | $v'_j$  | $v'_k$  | $v'_i$  |
| 15 | $v'_j$  | $v'_k$  | $v'_i$  |
| 16 | $v'_j$  | $v'_k$  | $v'_i$  |
| 17 | $v'_j$  | $v'_k$  | $v'_i$  |
| 18 | $v'_j$  | $v'_k$  | $v'_i$  |
| 19 | $v'_j$  | $v'_k$  | $v'_i$  |
| 20 | $v'_j$  | $v'_k$  | $v'_i$  |
| 21 | $v'_j$  | $v'_k$  | $v'_i$  |
| 22 | $v'_j$  | $v'_k$  | $v'_i$  |
| 23 | $v'_j$  | $v'_k$  | $v'_i$  |
| 24 | $v'_j$  | $v'_k$  | $v'_i$  |
| 25 | $v'_j$  | $v'_k$  | $v'_i$  |
| 26 | $v'_j$  | $v'_k$  | $v'_i$  |
| 27 | $v'_j$  | $v'_k$  | $v'_i$  |
| 28 | $v'_j$  | $v'_k$  | $v'_i$  |
| 29 | $v'_j$  | $v'_k$  | $v'_i$  |
| 30 | $v'_j$  | $v'_k$  | $v'_i$  |
| 31 | $v'_j$  | $v'_k$  | $v'_i$  |
| 32 | $v'_j$  | $v'_k$  | $v'_i$  |
| 33 | $v'_j$  | $v'_k$  | $v'_i$  |
| 34 | $v'_j$  | $v'_k$  | $v'_i$  |

4. If $j' < k' < j < k < i' < i$, then $(v'_j, v_i) \in A$, $(v'_i, v_i) \in A$, $(v'_j, v_j) \in A$, and $(v'_i, v_j) \in A$.

5. If $j' < k' < j < k < i' = i$, then $(v'_j, v_i) \in A$, $(v'_i, v_i) \in A$, $(v'_j, v_j) \in A$, and $(v'_i, v_j) \in A$.  

Springer
6. If \( j' < k' < j < k < i < i' \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, (v_{j'}, v_j) \in A, \) and \((v_{i'}, v_j) \in A. \\
7. If \( j' < j < k' < i' < k < i \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, (v_{j'}, v_j) \in A, \) and \((v_{i'}, v_j) \in A. \\
8. If \( j' < j < k' < k < i' < i \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, (v_{j'}, v_j) \in A. \\
But arc \((v_{i'}, v_j)\) does not belong to \( A \) because of the positions of \( i, i', j, j' \). Next we show that \((v_{i'}, v_j) \in A \) using \((v_i, v_{j'})\) which also does not belong to \( A \) because of the positions of \( i, i', j, j' \). If \((v_{i'}, v_j) \notin A \) and \((v_i, v_{j'}) \notin A \) then \( j' < j < k' < k < i' < i \) implies that \( co-G \) has an induced \( 2\overrightarrow{P}_2 \) which is not possible by our assumption. Thus we know that \((v_{i'}, v_j) \in A \) or \((v_i, v_{j'}) \in A. \\
If \((v_i, v_{j'}) \in A \) we can exchange the positions of \( k' \) and \( k \) (which implies that we are case 11) or we also have \((v_{i'}, v_j) \in A. \\
9. If \( j' < j < k' < k < i' = i \), then \((v_{j'}, v_i) \in A, (v_{j'}, v_j) \in A, \) and \((v_{i'}, v_j) \in A. \\
10. If \( j' < j < k' < k < i < i' \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, (v_{j'}, v_j) \in A, \) and \((v_{i'}, v_j) \in A \) (same situation as 8). \\
11. If \( j' < j < k < k' < i' < i \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, (v_{j'}, v_j) \in A. \\
But arc \((v_i, v_{j'})\) does not belong to \( A \) because of the positions of \( i, i', j, j' \). Next we show that \((v_i, v_{j'}) \in A \) using \((v_{i'}, v_j)\) which also does not belong to \( A \) because of the positions of \( i, i', j, j' \). If \((v_i, v_{j'}) \notin A \) and \((v_{i'}, v_j) \notin A \) then \( j' < j < k < k' < i' < i \) implies that \( co-G \) has an induced \( 2\overrightarrow{P}_2 \) which is not possible by our assumption. Thus we know that \((v_i, v_{j'}) \in A \) or \((v_{i'}, v_j) \in A \). If \((v_{i'}, v_j) \in A \) we can exchange the positions of \( k' \) and \( k \) (which implies that we are case 8) or we also have \((v_{i'}, v_j) \in A. \\
12. If \( j' < j < k < k' < i' = i \), then \((v_j, v_{i'}) \in A, (v_j, v_{j'}) \in A, \) and \((v_i, v_{j'}) \in A. \\
13. If \( j' < j < k < k' < i < i' \), then \((v_j, v_{i'}) \in A, (v_i, v_{i'}) \in A, (v_j, v_{j'}) \in A, \) and \((v_i, v_{j'}) \in A \) (same situation as 11). \\
14. If \( j' < j < k < i < k' < i' \), then \((v_j, v_{i'}) \in A, (v_i, v_{i'}) \in A, (v_j, v_{j'}) \in A, \) and \((v_i, v_{j'}) \in A. \\
15. If \( j' = j < k' < i' < k < i \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, \) and \((v_{i'}, v_j) \in A. \\
16. If \( j' = j < k' < k < i' < i \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, \) and \((v_{i'}, v_j) \in A. \\
17. If \( j' = j < k' < k < i < i' \), then \((v_{j'}, v_i) \in A, (v_{i'}, v_i) \in A, \) and \((v_{i'}, v_j) \in A. \\

Every remaining case \( i \in \{18, \ldots, 34\} \) is symmetric to shown case \( 35 - i \). Thus for every two missing arcs \((v_i, v_j) \in A - A' \) and \((v_{i'}, v_{j'}) \in A - A' \) which are inserted by additionally items all arcs between these items belong to \( A \).

Now we can show the claim by induction on the number \( m = |A - A'| \) of missing arcs. Let \( a_1, \ldots, a_m \) be an arbitrary order of the missing arcs. For \( m = 1 \) we can insert the missing arc by the assumption of the claim. Let \( m \geq 1, q_m \) be the sequence obtained from \( q' \) by inserting \( m \) missing arcs, and \( G_m = g(q_m) = (V, A_m) \). In order to insert arc \( a_{m+1} = (v_i, v_j) \) we neglect the \( 2m \) new items in \( q_m \) and consider only
the items of $q'$ for determining a position $k_{a_{m+1}}$, $j < k_{a_{m+1}} \leq i$, such that $a_{m+1}$ can be inserted into $G'$ by first inserting an additional item for type $v_j$ and then an additional item for type $v_i$ at position $k_{a_{m+1}}$. The existence of this position follows by the assumption of the claim. The insertion of the two new items at position $k_{a_{m+1}}$ for arc $a_{m+1} = (v_i, v_j)$ defines the following sets of arcs.

1. $A_{1,m+1} = \{(v_i, v_j)\}$
2. Set $A_{2,m+1}$ of all arcs from vertices corresponding to items of $q'$ on position $\leq k_{a_{m+1}}$ to $v_i$ and to $v_j$ and all arcs from $v_i$ and from $v_j$ to all vertices corresponding to items of $q'$ on position $> k_{a_{m+1}}$.
3. Set $A_{3,m+1}$ of all arcs from vertices corresponding to new items of $q_m$ on positions $k_{a_j}$, $j \leq m$, to $v_i$ and to $v_j$ and all arcs from $v_i$ and from $v_j$ to all vertices corresponding to new items of $q_m$ on positions $k_{a_j}$, $j \geq m$.

It remains to show that $A_{1,m+1} \cup A_{2,m+1} \cup A_{3,m+1} \subseteq A$. $A_{1,m+1} \subseteq A$ holds by definition and $A_{2,m+1} \subseteq A$ holds by the assumption of the claim. Set $A_{3,m+1}$ is equal to the union of all sets of arcs between the vertices corresponding to the two new inserted items for $a_{m+1}$ and the vertices corresponding to the two new inserted items for every $a_j$, $j \leq m$ following the order of the items. As our case distinction above implies that all arcs between the new items inserted at any $a_j$, $j \leq m$ and the new items inserted at $a_{m+1}$ are in $A$ the union of all these arcs $A_{3,m+1}$ is also in $A$.

So we can insert $a_{m+1}$ by first inserting an additional item for type $v_j$ and then an additional item for type $v_i$ at position $k'_{a_{m+1}}$, where $k'_{a_{m+1}}$ is the position in $q_m$ which corresponds to position $k_{a_{m+1}}$ in $q'$.

Thus we have shown that $g(\{q_m\}) = G \in S_{1,\ell}$ for some $\ell \geq 2$. \hfill \Box

Assume that $G \not\in S_{1,\ell}$ for every $\ell \geq 2$. By Claim 1 there is some arc $(v_i, v_j) \in A - A'$ such that for every position $k$, $j < k \leq i$ inserting an additional item for type $v_i$ and an additional item for type $v_j$ at position $k$ defines an arc which is not in $A$. That is, for every position $k$, $j < k \leq i$, in $q'$ there exists some $m'$, $j < m' \leq k$, such that it holds $(v_{m'}, v_j) \not\in A$ or there exists some $m''$, $k \leq m'' < i$, such that it holds $(v_i, v_{m''}) \not\in A$. By the transitivity of co-$G$ it follows that there is one position $k$, $j < k \leq i$, in $q'$ such that there exists some $m'$, $j < m' \leq k$, such that it holds $(v_{m'}, v_j) \not\in A$ and there exists some $m''$, $k \leq m'' < i$, such that it holds $(v_i, v_{m''}) \not\in A$.

If co-$G = (V, A^c)$ is the complement digraph of $G$ we know that

$$(v_{m'}, v_j) \in A^c \text{ and } (v_i, v_{m''}) \in A^c. \quad (3)$$

Since $m' \leq m''$ we know that $(v_{m'}, v_{m''}) \in A$. Further we know that $(v_{m''}, v_{m'}) \in A$, since otherwise $(v_{m''}, v_{m'}) \in A^c$, property (3), and the transitivity of co-$G$ would imply that $(v_i, v_j) \in A^c$ which is not possible. Thus we know that

$$(v_{m'}, v_{m''}) \not\in A^c \text{ and } (v_{m''}, v_{m'}) \not\in A^c. \quad (4)$$

Further the arcs $(v_j, v_{m'})$, $(v_j, v_{m''})$, $(v_{m'}, v_i)$, $(v_{m''}, v_i)$ belong to $A' \subseteq A$ and thus

$$(v_j, v_{m'}) \not\in A^c, \ (v_j, v_{m''}) \not\in A^c, \ (v_{m'}, v_i) \not\in A^c \text{ and } (v_{m''}, v_i) \not\in A^c. \quad (5)$$
If \((v_i, v_{m'}) \in A^c\) or \((v_{m''}, v_j) \in A^c\) then (3) and the transitivity of co-\(G\) would imply that \((v_i, v_j) \in A^c\), thus we know
\[
(v_i, v_{m'}) \notin A^c \text{ and } (v_{m''}, v_j) \notin A^c.
\]
Properties (3)–(6) imply that the subdigraph \((\{v_i, v_j, v_{m'}, v_{m''}\}, \{(v_i, v_{m''}), (v_{m'}, v_j)\})\) of co-\(G\) leads to a \(2\rightarrow P_2\), which implies that \(G\) contains a co-\((2P_2)\).

**Corollary 6** Every digraph in \(S_{k,\ell}\) can be obtained by the union of at most \(k\) many digraphs from \(S_{1,2}\) and thus by the union of at most \(k\) many \{co-(2P_2), 2K_1, C_3, D_4\}-free digraphs.

**Proposition 6** Let \(G \in S_{1,2}\), then a set \(Q\) of one sequence \(q\), such that \(G = g(Q)\) can be found in time \(O(|V| + |A|)\).

**Proof** Let \(G = (V, A) \in S_{1,2}\) and \(q = []\). We perform the following steps until \(G = (\emptyset, \emptyset)\).

- Choose \(v \in V\) such that \((v, u) \in A\) for all \(u \in V - \{v\}\) and append \(v\) to \(q\).
- Remove all arcs \((v, u)\) from \(A\).
- If indegree\((v) = \text{outdegree}(v) = 0\), remove \(v\) from \(V\).
- If there are vertices \(u\) such that indegree\((u) = \text{outdegree}(u) = 0\), remove \(u\) from \(V\) and append \(u\) to \(q\).

Next we show that for every \(G = (V, A) \in S_{1,2}\) this method leads to a sequence \(q\), such that \(G = g(|q|)\) and for every type in \(\text{types}(q)\) there are at most two items in \(q\).

In order to perform the algorithm there has to be an ordering \(v_1, \ldots, v_n\) of \(V\) such that for \(1 \leq i < n\) vertex \(v_i\) has maximum possible outdegree in subdigraph obtained by removing the outgoing arcs of \(v_1, \ldots, v_{i-1}\) and thereby created isolated vertices from \(G\). Since \(G \in S_{1,2}\) there is a sequence \(q'\) such that \(G = g(|q'|)\). The order in which the types corresponding to the vertices of \(V\) appear in subsequence \(F(q')\), defined in the proof of Theorem 3, ensures the existence of such an ordering.

By performing the operations of the algorithm along such a vertex ordering we observe that in every iteration only the outdegree of \(v\) is changed and only vertices with defined incoming and outgoing arcs are removed. Thus by the definition of the sequence digraph it holds \(G = g(|q|)\). Since every vertex which has only outgoing or only incoming arcs will be inserted once into \(q\) and every vertex which has outgoing and incoming arcs will be inserted at most twice into \(q\) this sequence fulfills the properties stated in the theorem.

**Example 5** We apply the method given in the proof of Proposition 6 on the digraph \(D_6\) in Table 2 and obtain the sequence
\[
q = [b, a, c, b, d, c].
\]
This even leads to a shorter sequence than we used in Example 4 (2).
4.3 Converse Sequence Digraphs and Complements of Sequence Digraphs

For some sequence \( q_i = (b_{i,1}, \ldots, b_{i,n_i}) \) we define by \( q_i^{-1} = (b_{n_i,i}, \ldots, b_{i,1}) \) its converse sequence and for some set \( Q \) we define by \( Q^{-1} = \{q_i^{-1} \mid q_i \in Q\} \) the set of converse sequences. Next we state a close relation between the sequence digraph of some set \( Q \) and of the converse digraph of the sequence digraph of \( Q^{-1} \).

**Lemma 18** Let \( Q \) be a set of sequences. Then it holds \( G = g(Q) \) if and only if \( con-G = g(Q^{-1}) \).

This relation implies that the classes \( S_{k,\ell} \) are closed under taking converse digraphs.

**Proposition 7** Let \( G \in S_{k,\ell} \), then \( con-G \in S_{k,\ell} \).

Next we want to address the relationship between the complement digraph of a sequence digraph \( g(Q) \) for some set \( Q \) and the sequence digraph \( g(Q^{-1}) \) of the converse sequences \( Q^{-1} \). In general these digraphs are different. This can be verified by the two sequence digraphs considered in Example 4. Under certain conditions for some set of sequences \( Q \) the complement digraph of \( g(Q) \) can be obtained by the sequence digraph of the converse sequences \( Q^{-1} \), which is useful since \( Q^{-1} \) has the same number of sequences and the same number of items per types as \( Q \).

**Lemma 19** Digraph \( G \) is a tournament if and only if \( co-G = con-G \).

**Proof** Digraph \( G \) is a tournament, if and only if between every two vertices of \( G \) there is exactly one arc, if and only if the converse digraph \( con-G \) corresponds to the complement digraph \( co-G \). \( \square \)

**Lemma 20** Let \( G \in S_{k,\ell} \) defined by some set of sequences \( Q \), such that graph \( g(Q) \) is a tournament, then \( co-(g(Q)) = g(Q^{-1}) \).

**Proof** Since \( g(Q) \) is a tournament by Lemma 19 it holds \( co-(g(Q)) = con-(g(Q)) \) and by Lemma 18 it holds \( con-(g(Q)) = g(Q^{-1}) \). \( \square \)

Since \( Q \) and \( Q^{-1} \) are defined on the same number of sequences and items of each type the following result holds.

**Corollary 7** Let \( G \in S_{k,\ell} \) be a tournament, then \( co-G \in S_{k,\ell} \).

Since the class \( S_{1,1} \) is equivalent to the class of transitive tournaments (Theorem 1), the last two results hold for digraphs in \( S_{1,1} \).

**Lemma 21** Let \( G \in S_{1,1} \) defined by some set \( Q = \{q_1\} \) of one sequence, then \( co-(g(Q)) = g(Q^{-1}) \).
Proposition 8 Let \( G \in S_{1,1} \), then \( \text{co-} G \in S_{1,1} \).

The results can not be generalized to graphs from \( S_{1,2} \) and also not sets \( Q \) on more than one sequence since these are not tournaments in general.

Beside transitivity shown in Lemma 13 we observe that the complements digraphs of sequence digraphs are special directed acyclic graphs.

Lemma 22 Let \( \ell \geq 1 \) and \( G \in S_{1,\ell} \), then the complement digraph \( \text{co-} G \) is acyclic.

Proof Let \( Q = \{q_1\} \) be some sequence which defines the sequence digraph \( g(Q) \). If the complement digraph \( \text{co-}(g(Q)) \) contains a cycle \( \{(v_1, \ldots, v_n), \{(v_1, v_2), \ldots, (v_{n-1}, v_n), (v_n, v_1)\}\}, n \geq 2 \), then by Lemma 13 vertex set \( \{v_1, \ldots, v_n\} \) induces a bidirectional complete subdigraph in digraph \( \text{co-}(g(Q)) \). Thus the vertex set \( \{v_1, \ldots, v_n\} \) induces an edgeless subdigraph in \( g(Q) \) which contradicts Lemma 7.

It is well known that every DAG has at least one sink, i.e. a vertex \( v_1 \) of outdegree 0, and one source, i.e. a vertex \( v_2 \) of indegree 0 by [7]. This means by Lemma 22 that for every \( \ell \geq 1 \) every digraph \( G = (V, A) \in S_{1,\ell} \) has a vertex \( v_1 \) of outdegree \( |V| - 1 \) and a vertex \( v_2 \) of indegree \( |V| - 1 \). For \( \ell = 1 \) this is known by Theorem 1.

Lemma 23 Let \( \ell \geq 1 \) and \( G = (V, A) \in S_{1,\ell} \), \( u \) and \( v \) be two vertices from \( V \), then the complement digraph \( \text{co-} G \) has at most one of the two arcs \( (u, v) \) and \( (v, u) \), i.e. \( \text{co-} G \) is an oriented graph.

Proof Let \( G = (V, A) \in S_{1,\ell} \), \( u \) and \( v \) be two distinct vertices from \( V \). Then \( G \) contains at least one of the two arcs \( (u, v) \) and \( (v, u) \) which implies that \( \text{co-} G \) has at most one of the two arcs \( (u, v) \) and \( (v, u) \).

Proposition 9 Let \( \ell \geq 1 \) and \( G \in S_{1,\ell} \), then the complement digraph \( \text{co-} G \) has at most one non-trivial component which contains arcs.

Proof If we assume that \( \text{co-} G \) has more than one non-trivial component which contains arcs, then by Lemma 23 digraph \( \text{co-} G \) contains \( 2\overrightarrow{P}_2 = \{(a, b, c, d), ((c, a), (d, b))\} \), see Table 2, as an induced subdigraph. Lemma 14 leads to a contradiction.

Digraphs in \( S_{2,1} \) consist of the disjoint union of two digraphs in \( S_{1,1} \) (see Lemma 12), thus the corresponding complements have exactly one component. The complements of digraphs in \( S_{2,2} \) can have more than one non-trivial component which contains arcs. This can be verified by digraph \( \text{co-} G \in S_{2,2} \) in the following example, whose complement \( G \) has two non-trivial components which contains arcs.
Example 6 Let $G$ be the disjoint union of two digraphs $G' = (V', A')$ and $G'' = (V'', A'')$ from $S_{1, 1}$. Then $G$ can be defined by $Q = \{q_1, q_2\}$, where $q_1 = [v'_1, v'_2, \ldots, v'_n]$ and $q_2 = [v''_1, v''_2, \ldots, v''_m]$. W.l.o.g. we assume the vertices in $V'$ and $V''$ to be enumerated in the order in which the graphs can be defined by inserting in-dominated vertices (cf. Theorem 1. (8)). Thus it holds $G \in S_{2, 1}$.

The complement digraph $co-G$ can be defined as follows. The digraph complement of a digraph in $S_{1, 1}$ also belongs to $S_{1, 1}$ and the additional arcs between $co-G'$ and $co-G''$ can be defined by two sequences, one for the arcs from $co-G'$ to $co-G''$ and one for the arcs from $co-G''$ to $co-G'$. The arcs within $co-G'$ and $co-G''$ can be defined in both sequences. Formally $co-G$ can be defined by $Q' = \{q'_1, q'_2\}$, where

$$q'_1 = [v'_n, v'_{n-1}, \ldots, v'_1, v''_m, v''_{m-1}, \ldots, v'_1]$$

and

$$q'_2 = [v''_m, v''_{m-1}, \ldots, v''_1, v'_n, v'_{n-1}, \ldots, v'_1].$$

Thus it holds $co-G \in S_{2, 2}$.

Next we want to study for some digraph $G \in S_{k, \ell}$ whether it holds $co-G \in S_{k, \ell}$. By Proposition 8 the class $S_{1, 1}$ is closed under complementations. The next example shows that the class $S_{1, 2}$ is not closed under complementations and that the values of $k$ and $\ell$ can grow arbitrary.

Example 7 We consider

$$\overrightarrow{K_{n,m}} = ([v_1, \ldots, v_n, w_1, \ldots, w_m], \{(v_i, w_j) \mid 1 \leq i \leq n, 1 \leq j \leq m\})$$

as an orientation of a complete bipartite graph $K_{n,m}$. The complement digraph of $\overrightarrow{K_{n,m}}$ can be defined by one sequence

$$q = [w_1, w_2, \ldots, w_m, w_1, w_2, \ldots, w_m, v_1, v_2, \ldots, v_n]$$

which implies that the complement digraph of $\overrightarrow{K_{n,m}}$ is in $S_{1, 2}$. By Lemma 9 it holds $\overrightarrow{K_{n,m}} \in S_{n,m,\max(n,m)}$ but $\overrightarrow{K_{n,m}} \notin S_{k,\ell}$ for $k < n \cdot m$ or $\ell < \max(n, m)$.

Since $S_{1, 2} \subseteq S_{k, \ell}$ for $k \geq 1$ and $\ell \geq 2$ we know that the classes $S_{k, \ell}$ for every $k \geq 1$ and $\ell \geq 2$ are not closed under complementations. The next examples show that $S_{2, 1}$ is not closed under complementations.

Example 8 1. Let $G_1$ be the disjoint union of two 1-vertex digraphs in $S_{1, 1}$. Then $G_1 \in S_{2, 1}$. Further $co-G_1 = \overrightarrow{K_2} \in S_{1, 2} - S_{2, 1}$.

2. Let $G_n$ for $n \geq 2$ be the disjoint union of two $n$-vertex digraphs in $S_{1, 1}$. Then $G_n \in S_{2, 1}$ and $co-G_n \in S_{2, 2} - (S_{2, 1} \cup S_{1, 2})$, see Example 6.

Lemma 24 Let $G \in S_{k, 1}$ for $k \geq 1$, then $co-G \in S_{2, 2}$.

Proof By Lemma 12 every digraph $G \in S_{k, 1}$ is the disjoint union of $k$ digraphs $G_1, \ldots, G_k$ from $S_{1, 1}$. Let $q_i$ be the sequence for $G_i$. For every $i$ it holds $co-G_i \in S_{2, 2}$.
S_{1,1} (Proposition 8). A sequence for co-\(G_i\) can be obtained by the reverse sequence \(q_i^{-1}\) of that for \(G_i\) (Lemma 21). Thus digraph co-\(G\) can be obtained by \(k\) digraphs from \(S_{1,1}\) and all possible arcs between two of these \(k\) graphs. The arcs between these \(k\) digraphs can be created by two sequences each consisting of concatenations of \(q_i^{-1}\) as follows.

\[
q' = q_1^{-1} \circ q_2^{-1} \circ \ldots \circ q_k^{-1}
\]

\[
q'' = q_k^{-1} \circ q_{k-1}^{-1} \circ \ldots \circ q_1^{-1}
\]

Since co-\(G = g(\{q', q''\})\) it holds that co-\(G \in S_{2,2}\).

\[\square\]

### 4.4 Simple Sequences

Let \(Q = \{q_1, \ldots, q_k\}\) be a set of \(k\) sequences. By Observation 2 only the first and the last item of each type in every \(q_i \in Q\) are important for the arcs in the corresponding digraph. Next we want to analyze how we even can use subsequences defined by the first or last item of each type. Let \(F(q_i)\) be the subsequence of \(q_i\) which is obtained from \(q_i\) by removing all except the first item for each type and \(L(q_i)\) be the subsequence of \(q_i\) which is obtained from \(q_i\) by removing all except the last item for each type.\(^5\)

**Example 9** \((F(q_i), L(q_i))\) For the sequence \(q_1 = [b, a, c, b, a]\) we obtain the two subsequences \(F(q_1) = [b, a, c]\) and \(L(q_1) = [c, b, a]\).

The given example shows that sequence digraph \(g(\{q_1\})\) differs from the union of the two sequence digraphs \(g(\{F(q_1)\})\) and \(g(\{L(q_1)\})\) only by arc \((a, b)\). In order to formalize this difference we define special sequences. Sequence \(q_i \in Q\) is simple, if there do not exist \(t, t' \in \text{types}(q_i)\) such that

\[
\text{first}(q_i, t) < \text{first}(q_i, t') < \text{last}(q_i, t) < \text{last}(q_i, t').
\]  

(7)

Set \(Q\) is simple, if every sequence \(q_i \in Q\) is simple. We define \(S'_{k, \ell}\) to be the set of all sequence digraphs defined by simple sets \(Q\) on at most \(k\) sequences that contain at most \(\ell\) items of each type in \(\text{types}(Q)\). By the definition we know for every two integers \(k \geq 1\) and \(\ell \geq 1\) the following inclusions between these graph classes.

\[
S'_{k, \ell} \subseteq S_{k, \ell}
\]

(8)

\[
S'_{k, 1} = S_{k, 1}
\]

(9)

The sequence \(q_1 = [b, a, c, c, b, a]\) given in Example 9 is not simple and sequence digraph \(g(\{q_1\})\) differs from the union of the two sequence digraphs \(g(\{F(q_1)\})\) and \(g(\{L(q_1)\})\). The next theorem says that this is no coincidence.

**Theorem 4** A sequence \(q_1\) is simple if and only if \(g(\{q_1\})\) can be obtained by the union of \(g(\{F(q_1)\})\) and \(g(\{L(q_1)\})\).

\[^5\]If for some type there is only one item in \(q_i\), then this item remains in \(F(q_i)\) and in \(L(q_i)\).
Proof (⇒) Let \( q_1 \) be a simple sequence and \( g(\{q_1\}) = (V, A) \) its sequence digraph. Further let \( g(\{F(q_1)\}) = (V, A') \) and \( g(\{L(q_1)\}) = (V, A'') \) be the sequence digraphs of \( F(q_1) \) and \( L(q_1) \). We have to show that \( A = A' \cup A'' \).

To show \( A \subseteq A' \cup A'' \) we consider subsequence \( q'_1 = M(q_1) \) (defined in Section 3.1), which is also simple, defines the same digraph as \( q_1 \), and contains at most two items for every type. This implies that for every arc \((u, v)\in A \) vertices \( u \) and \( v \) correspond to the first or the last item of its type.

- If \((u, v) \in A \), first\(q'_1, u) = b_i \), and first\(q'_1, v) = b_j \) for some \( i < j \), then \((u, v) \in A' \).
- If \((u, v) \in A \), last\(q'_1, u) = b_i \), and last\(q'_1, v) = b_j \) for some \( i < j \), then \((u, v) \in A'' \).
- If \((u, v) \in A \), last\(q'_1, u) = b_i \), and first\(q'_1, v) = b_j \) for some \( i < j \), then there is some \( i' < i \) such that first\(q_1, u) = b_{i'} \), which implies that \((u, v) \in A' \). Further there is also some \( j' \geq j \) such that last\(q_1, v) = b_{j'} \), which implies that \((u, v) \in A'' \).
- If \((u, v) \in A \), first\(q'_1, u) = b_i \), and last\(q'_1, v) = b_j \) for some \( i < j \) we consider the following cases. If there is also some \( i', i' < j \), such that last\(q'_1, u) = b_{i'} \), then \((u, v) \in A'' \). If there is also some \( j', i < j' \leq j \), such that first\(q'_1, v) = b_{j'} \), then \((u, v) \in A' \). If none of the two cases is fulfilled, then first\(q'_1, v) < first(q'_1, u) < last(q'_1, v) < last(q'_1, u) \), which is not possible within a simple sequence.

Further by the definition \( g(\{F(q_1)\}) \) and \( g(\{L(q_1)\}) \) are subgraphs of \( G \) and thus it holds that \( A' \cup A'' \subseteq A \).

(⇐) If \( q_1 \) is not simple, there exist \( t, t' \in \text{types}(q_1) \) such that (7) holds for \( i = 1 \). Thus arc \((t', t) \) is in \( g(\{q_1\}) \) but not in \( g(\{F(q_1)\}) \) or \( g(\{L(q_1)\}) \), which implies that \( g(\{q_1\}) \) can not be obtained by the union of \( g(\{F(q_1)\}) \) and \( g(\{L(q_1)\}) \). \( \square \)

By the definition for every sequence \( q_i \) the digraphs \( g(\{F(q_i)\}) \) and \( g(\{L(q_i)\}) \) both are in \( S_{1,1} \) (see Theorem 1 for a precise characterization).

**Corollary 8** Let \( G \in S'_{k,\ell} \), then \( G \) can be obtained by the union of 2k digraphs from \( S_{1,1} \).

In order to generalize this result to arbitrary sequences it remains to add the missing arcs \((i', t) \) for every \( t, t' \in \text{types}(q_1) \) such that (7) holds by using further sequences. This is not possible within one additionally sequence for each non-simple sequence.

### 5 Directed Path-width of Sequence Digraphs

#### 5.1 Directed Path-Width

According to Barát [11], the notion of directed path-width was introduced by Reed, Seymour, and Thomas around 1995 and relates to directed tree-width introduced by
Johnson, Robertson, Seymour, and Thomas in [12]. A directed path-decomposition of a digraph \( G = (V, A) \) is a sequence \( (X_1, \ldots, X_r) \) of subsets of \( V \), called bags, such that the following three conditions hold true.

1. (dpw-1) \( X_1 \cup \ldots \cup X_r = V \).
2. (dpw-2) For each \((u, v) \in A\) there is a pair \(i \leq j\) such that \( u \in X_i \) and \( v \in X_j \).
3. (dpw-3) If \( u \in X_i \) and \( u \in X_j \) for some \( u \in V \) and two indices \( i, j \) with \( i \leq j \), then \( u \in X_\ell \) for all indices \( \ell \) with \( i \leq \ell \leq j \).

The width of a directed path-decomposition \( X = (X_1, \ldots, X_r) \) is

\[
\max_{1 \leq i \leq r} |X_i| - 1.
\]

The directed path-width of \( G \), \( d\text{-pw}(G) \) for short, is the smallest integer \( w \) such that there is a directed path-decomposition for \( G \) of width \( w \).

There is a close relation between the directed path-width of some digraph and the (undirected) path-width (see [13]) of its underlying undirected graph.

**Lemma 25** (Lemma 1 of [11]) Let \( G \) be some complete bioriented digraph, then \( d\text{-pw}(G) = \text{pw(und}(G)) \).

Determining whether the (undirected) path-width of some given (undirected) graph is at most some given value \( w \) is NP-complete [14] even for bipartite graphs, complements of bipartite graphs [15], chordal graphs [16], bipartite distance hereditary graphs [17], and planar graphs with maximum vertex degree 3 [18]. Lemma 25 implies that determining whether the directed path-width of some given digraph is at most some given value \( w \) is NP-complete even for digraphs whose underlying graphs lie in the mentioned classes. For example determining whether the directed path-width of some given digraph with maximum semi-degree \( \Delta^0(G) = \max(\Delta^-(D), \Delta^+(D)) \leq 3 \) is at most some given value \( w \) is NP-complete, which will be useful in Proposition 11. While undirected path-width can be solved by an FPT-algorithm [19], the existence of such an algorithm for directed path-width is still open. The directed path-width of a digraph \( G = (V, A) \) can be computed in time \( \mathcal{O}(|A| \cdot |V|^{2d\text{-pw}(G)}/(d\text{-pw}(G) - 1)!)) \) by [3] and in time \( \mathcal{O}(d\text{-pw}(G) \cdot |A| \cdot |V|^{2d\text{-pw}(G)}) \) by [4]. This leads to XP-algorithms for directed path-width w.r.t. the standard parameter and implies that for each constant \( w \), it is decidable in polynomial time whether a given digraph has directed path-width at most \( w \). Further in [5] it is shown how to decide whether the directed path-width of an \( \ell \)-semicomplete digraph is at most \( w \) in time \((\ell + 2w + 1)^{2w} \cdot |V|^{\mathcal{O}(1)}\).

### 5.2 Hardness of Directed Path-width on Sequence Digraphs

Next we give some conditions on the sequences in \( Q \) such that for the corresponding digraph \( g(Q) \) computing its directed path-width is NP-hard.
Proposition 10 Given some set \( Q \) on \( k \) sequences such that \( n_i = 2 \) for \( 1 \leq i \leq k \) and some integer \( p \), then the problem of deciding whether \( d\text{-pw}(g(Q)) \leq p \) is NP-complete.

Proof The stated problem is in NP since we can restrict to directed path-decompositions having a polynomial number of bags. To show the NP-hardness by a reduction from the directed path-width problem we transform instance \((G, p)\) in linear time into instance \((q(G), p)\) for the stated problem. The correctness follows by Observation 3.

Proposition 11 Given some set \( Q \) such that \( d_Q = 3 \) or \( c_Q = 5 \) and some integer \( p \), then the problem of deciding whether \( d\text{-pw}(g(Q)) \leq p \) is NP-complete.

Proof We show the NP-hardness by a reduction from the path-width problem for undirected graphs \( G \) where \( \Delta(G) \leq 3 \). Then the complete biorientation \( \overrightarrow{G} \) fulfills \( \max(\Delta^-(\overrightarrow{G}), \Delta^+(\overrightarrow{G})) \leq 3 \) and by Lemma 25 it holds \( d\text{-pw}(\overrightarrow{G}) = \text{pw}(G) \). The instance for the stated problem is \((Q, p)\), where \( Q \) can be obtained from \( q(\overrightarrow{G}) \) by Lemma 6. The correctness follows by Observation 3.

Similar results can be shown for related parameters such as directed cut-width [20].

5.3 Polynomial Cases of Directed Path-Width on Sequence Digraphs

Next we consider the directed path-width of sequence digraphs for \( k = 1 \) or \( \ell = 1 \).

Proposition 12 Let \( G \in S_{k,1} \), then \( d\text{-pw}(G) = 0 \).

Proof By Theorem 2 every digraph in \( S_{k,1} \) is the disjoint union of \( k \) digraphs in \( S_{1,1} \). By Theorem 1 every digraph in \( S_{1,1} \) is acyclic and thus has directed path-width 0.

On the other hand, there are no constants \( k, \ell \) such that every digraph of directed path-width 0 is in \( S_{k,\ell} \). This can be verified by an orientation \( T' \) of a tree \( T \) on \( k \) edges and \( \Delta(T) = \ell \). Then \( d\text{-pw}(T') = 0 \) and by Lemma 9 it holds \( T' \in S_{k,\ell} \) but \( T' \notin S_{k',\ell'} \) for \( k' < k \) or \( \ell' < \ell \).

For digraphs in \( S_{1,2} \) the directed path-width can be arbitrary large, since this class includes all bidirectional complete digraphs. We can compute this value as follows. Let \( Q = \{q\} \). For type \( t \in \text{types}(q) \) let \( I_t = [\text{first}(q,t), \text{last}(q,t)] \) be the interval representing \( t \), and let \( I_q = \{I_t \mid t \in \text{types}(q)\} \) be the set of all intervals for sequence \( q \). Let \( I(q) = (V, E) \) be the interval graph where \( V = \text{types}(q) \) and \( E = \{\{u, v\} \mid u \neq v, I_u \cap I_v \neq \emptyset, I_u, I_v \in I_q\} \), see Fig. 3.

Proposition 13 Let \( G \in S_{1,2} \) defined by some set \( Q = \{q_1\} \) of one sequence, then \( d\text{-pw}(G) = \omega(I(q)) - 1 = \text{pw}(I(q)) \).
Proof. It holds \( \text{d-pw}(G) \leq \omega(I(q)) - 1 \) by an obvious directed path-decomposition along \( I(q) \). Further for every integer \( r \) the set \( I(r) = \{ I_t \mid r \in I_t \} \) defines a complete subgraph \( K_{|I(r)|} \) in \( I(q) \) and also a bidirectional complete subdigraph \( K_{|I(r)|} \) in \( G \). Thus it holds \( \text{d-pw}(G) \geq \omega(I(q)) - 1 \). The second equality holds since the (undirected) path-width of an interval graph is equal to the size of a maximum clique \([21]\). \( \square \)

In contrast to Proposition 11 sets \( Q \) where \( d_Q = 1 \) can be handled in polynomial time.

Proposition 14. Given some set \( Q \) such that \( d_Q = 1 \) and some integer \( p \), then the problem of deciding whether \( \text{d-pw}(g(Q)) \leq p \) can be solved in time \( O(|\text{types}(Q)|^2 + n) \).

Proof. Let \( Q = \{ q_1, \ldots, q_k \} \). If \( d_Q = 1 \) the vertex sets \( V_i = \text{types}(q_i) \) are disjoint. That is, \( g(Q) \) is the disjoint union of digraphs in \( S_{1,2} \) for which the directed path-width can be computed in time \( O(\sum_{i=1}^{k}|\text{types}(q_i)|^2 + n_i) \leq O(|\text{types}(Q)|^2 + n) \) by Proposition 13. \( \square \)

5.4 An XP-Algorithm for Directed Path-Width

We next give an XP-algorithm for directed path-width w.r.t. the parameter \( k \), which implies that for every constant \( k \) for a given set \( Q \) on at most \( k \) sequences the value \( \text{d-pw}(g(Q)) \) can be computed in polynomial time. The main idea is to discover an optimal directed path-decomposition by scanning the \( k \) sequences left-to-right and keeping in a state the numbers of scanned items of every sequence and a certain number of active types.

Since \( \text{d-pw}(G) = 0 \) for \( G \in S_{k,1} \) by Proposition 12, we assume that there are at least two items for each type in \( Q \). If this is not the case, we insert another item directly after the only item of some type for each type and get \( Q' \). Obviously, this does not change the sequence digraph, so it holds \( g(Q) = g(Q') \). Such an insertion of additional items does not change the asymptotic runtime, since \( |Q'| \leq 2 \cdot |Q| \). If algorithm \( A \) has runtime \( T_A(|I|) \in O(|I|^c) \) for some instance \( I \) and constant \( c \), then after the modification of the instance \( A \) has runtime \( T_A(|I'|) \leq T_A(2 \cdot |I|) \in O((2 \cdot |I|)^c) = O(2^c \cdot |I|^c) = O(|I|^c) \) since \( 2^c \) is constant.

Example 10. Let \( Q = \{ q_1, q_2 \} \) be the set of sequences \( q_1 = [a, b, c, b, d, a] \) and \( q_2 = [e, f, e, c] \). Then there is only one item for type \( d \) and one item for \( f \) contained in the sequences, so we add two further items, one for type \( d \) in \( q_1 \) and one for type \( f \) in \( q_2 \).
and we get the modified set of sequences \( Q' = \{q'_1, q'_2\} \) and \( q'_1 = [a, b, c, b, d, d, a] \) and \( q'_2 = [e, f, f, e, c] \). The sequence digraph \( g(Q) = g(Q') \) is given in Fig. 4.

Let \( Q = \{q_1, \ldots, q_k\} \) be a set of \( k \) sequences. Every \( k \)-tuple \((i_1, \ldots, i_k)\) where \( 0 \leq i_j \leq n_j \) for \( 1 \leq j \leq k \) is a state of \( Q \). State \((0, 0, \ldots, 0)\) is the initial state and \((n_1, \ldots, n_k)\) is the final state. The state digraph \( s(Q) \) for a set \( Q \) has a vertex for each possible state. There is an arc from vertex \( u \) labeled by \((u_1, \ldots, u_k)\) to vertex \( v \) labeled by \((v_1, \ldots, v_k)\) if and only if \( u_i = v_i - 1 \) for exactly one element of the vector and for all other elements of the vector \( u_j = v_j \).

Let \( s = (i_1, \ldots, i_k) \) be a state of \( Q \). We define \( L(s) \) to be the set of all items on the positions \( 1, \ldots, i_j \) for \( 1 \leq j \leq k \) and \( R(s) \) is the set of all items on the remaining positions \( i_j + 1, \ldots, n_j \) for \( 1 \leq j \leq k \). Furthermore, we define \( M(s) \) to be the set of items on the positions \( i_j + 1 \) for \( 1 \leq j \leq k \) and \( \text{front}(s) = \{t(b) \mid b \in M(s)\} \) to be the set of types of the items in \( M(s) \). Obviously, for every state \( s = (i_1, \ldots, i_k) \) it holds that \( L(s) \cup R(s) \) leads to a disjoint partition of the items in \( Q \) and \( M(s) \subseteq R(s) \).

Further each vertex \( v \) of the state digraph is labeled by the value \( f(v) \). This value is the number of types \( t \) such that there is at least one item of type \( t \) in \( L(v) \) and at least one item of type \( t \) in \( R(v) \). Formally we define \( \text{active}(v) = \{t \in \text{types}(Q) \mid b \in L(v), t(b) = t, b' \in R(v), t(b') = t\} \) and \( f(v) = |\text{active}(v)| \). Obviously, for the initial state \( u \) and the final state \( v \) it holds \( |\text{active}(u)| = |\text{active}(v)| = 0 \). Since the state digraph \( s(Q) \) is a directed acyclic graph we can compute all values \( |\text{active}(v)| \) using a topological ordering \( \text{topol} \) of the vertices. Every arc \((u, v)\) in \( s(Q) \) represents one item \( b_{i,j} \), thus

\[
|\text{active}((i_1, \ldots, i_{j-1}, i_j + 1, i_{j+1}, \ldots, i_k))|
= |\text{active}((i_1, \ldots, i_{j-1}, i_j, i_{j+1}, \ldots, i_k))| + c_j
\]

where

\[
c_j = \begin{cases} 
1, & \text{if } \text{first}(q_j, t) = i_j + 1 \text{ and } \text{first}(q_\ell, t) > i_\ell \ \forall \ \ell \neq j \\
-1, & \text{if } \text{last}(q_j, t) = i_j + 1 \text{ and } \text{last}(q_\ell, t) \leq i_\ell \ \forall \ \ell \neq j \\
0, & \text{otherwise}.
\end{cases}
\]

Please remember our technical definition of \( \text{first}(q, t) \) and \( \text{last}(q, t) \) from Section 3.1 for the case that \( t \notin \text{types}(q) \). Thus, the calculation of value \( |\text{active}(i_1, \ldots, i_k)| \) for

![Fig. 4 Sequence digraph of Example 10](https://example.com/fig4.png)
the vertex labeled \((i_1, \ldots, i_k)\) depends only on already calculated values, which is necessary in order to use dynamic programming.

**Example 11** (State digraph) Let \(Q = \{q_1, q_2\}\) be the set of sequences \(q_1 = [a, b, c, b, d, d, a]\) and \(q_2 = [e, f, e, c]\). The state digraph \(s(Q)\) is given in Fig. 5. For every vertex \(v\) of \(s(Q)\) the vector \(h(v)\) in the vertex represents the state of \(v\) and every arc is labeled with the type of the item, that is considered in order to obtain the next state, and the number \(f(v)\) is given on the left of the vertex that represents state \(h(v)\).

Let \(\mathcal{P}(Q)\) be the set of all paths from the initial state to the final state in \(s(Q)\). Every path \(P \in \mathcal{P}(Q)\) has \(r = 1 + \sum_{i=1}^{k} n_i\) vertices, i.e., \(P = (v_0, \ldots, v_r)\). First we observe that every path in \(\mathcal{P}(Q)\) leads to a directed path-decomposition for \(g(Q)\).

**Lemma 26** Let \(Q\) be some set of \(k\) sequences and \((v_0, \ldots, v_r) \in \mathcal{P}(Q)\). Then the sequence \((\text{active}(v_1), \ldots, \text{active}(v_{r-1}))\) is a directed path-decomposition for \(g(Q)\).
Proof Let \( g(Q) = (V, A) \) be the sequence digraph, \( s(Q) = (S, B) \) be the state digraph, and \( P = (v_0, \ldots, v_r) \) be an arbitrary path from the initial state to the final state in \( s(Q) \). We next will verify the three conditions of a directed path-decomposition. Since \( \text{types}(Q) = V \) we identify the vertices with their types.

- For any type \( t \in \text{types}(Q) \), let \( (v_i, v_j) \) be the first arc on the path \( P \) labeled by \( t \). Then \( t \in \text{active}(v_i) \) is valid, because later there is another arc \( (v_{j-1}, v_j) \) with \( i < j \), which is also marked with \( t \), because there are always at least two items per type. It follows that \( V = \text{active}(v_1) \cup \ldots \cup \text{active}(v_{r-1}) \).

- Let \( (t, t') \in A \). By the definition of \( g(Q) \) there is one sequence \( q_\ell \) such that there is an item \( b_{\ell,m} \) of type \( t \) and an item \( b_{\ell,m'} \) of type \( t' \) such that \( m < m' \). Thus there is a vertex \( v_i \) in \( P \) such that \( b_{\ell,m} \in L(v_i) - L(v_{i-1}) \) and a vertex \( v_j \) in \( P \) such that \( b_{\ell,m'} \in L(v_j) - L(v_{j-1}) \), and \( i < j \). If \( (v_{i-1}, v_i) \) is the first arc on the path \( P \) labeled by \( t \) then \( t \in \text{active}(v_i) \), else \( t \in \text{active}(v_{i-1}) \). If \( (v_{j-1}, v_j) \) is the first arc on the path \( P \) labeled by \( t' \) then \( t' \in \text{active}(v_j) \), else \( t' \in \text{active}(v_{j-1}) \). Thus \( t \in \text{active}(v_i) \) and \( t' \in \text{active}(v_j') \) for some \( i' \leq j' \).

- Let \( (v_{i-1}, v_i) \) be the first arc on path \( P \) labeled by \( t \) and let \( (v_k, v_{k+1}) \) be the last arc on path \( P \) labeled by \( t \). Then \( t \in \text{active}(v_j) \) is valid for \( i \leq j \leq k \), and \( t \not\in \text{active}(v_j) \) is valid for \( j < i \) or \( j > k \). It follows that for each \( t \in \text{active}(v_i) \cap \text{active}(v_k) \) it also holds \( t \in \text{active}(v_j) \) for \( i \leq j \leq k \).

Thus, sequence \( (\text{active}(v_1), \ldots, \text{active}(v_{r-1})) \) is a directed path-decomposition for \( g(Q) \).

Example 12 (Paths in \( s(Q) \)) We consider two paths in \( s(Q) \) of Example 11.

1. The path

\[
P_1 = ((0, 0), (0, 1), (0, 2), (1, 2), (2, 2), (3, 2), (3, 3), (3, 4), (3, 5), (4, 5), (5, 5), (6, 5), (7, 5))
\]

leads to the directed path-decomposition

\[
([e], [e, f], [a, e, f], [a, b, e, f], [a, b, c, e, f], [a, b, c, e],
[a, b, c], [a, b], [a], [a, d], [a])
\]

of width 4 for \( g(Q) \). The path is shown in red in Fig. 5.

2. The path

\[
P_2 = ((0, 0), (1, 0), (1, 1), (1, 2), (1, 3), (1, 4), (2, 4),
(3, 4), (4, 4), (5, 4), (6, 4), (7, 4), (7, 5))
\]

leads to the directed path-decomposition

\[
([a], [a, e], [a, e, f], [a, e], [a], [a, b], [a, b, c], [a, c], [a, c, d], [a, c], [c])
\]

of width 2 for \( g(Q) \). The path is shown in blue in Fig. 5.

Example 12 shows that \( P \in \mathcal{P}(Q) \) can lead to directed path-decompositions of non-optimal width. Lemma 26 leads to an upper bound on the directed path-width of \( g(Q) \) using the state digraph. The reverse direction is considered next.
Lemma 27 Let $Q$ be some set of $k$ sequences. If there is a directed path-decomposition $X = (X_1, \ldots, X_m)$ of width $p - 1$ for $g(Q)$, then there is a path $(v_0, \ldots, v_r) \in \mathcal{P}(Q)$ such that for every $1 \leq i \leq r$ it holds $|\text{active}(v_i)| \leq p$.

Proof For a type $t$ let $\alpha(X, t)$ be the smallest $i$ such that $t \in X_i$ and $\beta(X, t)$ be the largest $i$ such that $t \in X_i$, see Example 13. Then $t \in X_i$ if and only if $\alpha(X, t) \leq i \leq \beta(X, t)$. If $(t_1, t_2)$ is an arc of $g(Q)$, then $\alpha(X, t_1) \leq \beta(X, t_2)$. This follows by (dpw-2) of the definition of a directed path-decomposition.

We define the path $P = (u_0, \ldots, u_r)$ inductive in the following way. Consider the last vertex $u$ on the already constructed path. Let $v_1, \ldots, v_k$ be the neighbors of $u$ in state digraph $s(Q)$ and let $t_j$ be the type of the item that is considered in order to obtain vertex $v_j$.

(a) If there is a type $t \in \text{front}(u)$ such that $t \in \text{active}(u)$ then let $v_j$ be the next vertex on path $P$, if $t = t_j$.

(b) Otherwise, if there is no type $t \in \text{front}(u)$ such that $t \in \text{active}(u)$, then let $t \in \text{front}(u)$ be the type such that $\alpha(X, t)$ is minimal.

We next show that for every vertex $u$ of path $P$ there is a bag $X_i$ in the directed path-decomposition $X = (X_1, \ldots, X_m)$, such that $\text{active}(u) \subseteq X_i$. This implies that $|\text{active}(u)| \leq p$ holds for each vertex $u$ on path $P$.

First, let $v$ be a vertex we get by case (b) from above, let $u$ be the preceding vertex of $v$ on path $P$, and let arc $(u, v)$ be labeled with $t$ in state digraph $s(Q)$. Then $\alpha(X, t)$ is minimal, and let $t' \in \text{active}(u)$. We show that the intervals $[\alpha(X, t), \beta(X, t)]$ and $[\alpha(X, t'), \beta(X, t')]$ overlap, because $\alpha(X, t) \leq \beta(X, t')$ and $\alpha(X, t') \leq \beta(X, t)$.

(1) To show $\alpha(X, t) \leq \beta(X, t')$ we observe the following:

Since $t' \notin \text{front}(u)$, there has to be a type $t'' \in \text{front}(u)$ such that $(t'', t')$ is an arc of $g(Q)$. This implies that $\alpha(X, t'') \leq \beta(X, t')$. Since $t$ is a type of $\text{front}(u)$ for which $\alpha(X, t)$ is minimal, we have $\alpha(X, t) \leq \alpha(X, t'')$ and thus $\alpha(X, t) \leq \beta(X, t)$.

(2) To show $\alpha(X, t') \leq \beta(X, t)$ we observe the following:

Let $w'$ be the first vertex such that $t' \in \text{active}(w')$, and let $w$ be the preceding vertex of $w'$ on path $P$. We distinguish two cases:

(a) $t \in \text{front}(w)$. Since $t'$ is a type of $\text{front}(w)$ for which $\alpha(X, t')$ is minimal, we have $\alpha(X, t') \leq \alpha(X, t)$. Since $t$ is a type of $\text{front}(w)$, we have $\alpha(X, t) \leq \beta(X, t)$ and thus $\alpha(X, t') \leq \beta(X, t)$.

(b) $t \notin \text{front}(w)$. Then there is a type $t'' \in \text{front}(w)$ such that $(t'', t)$ is an arc of $g(Q)$. This implies that $\alpha(X, t'') \leq \beta(X, t')$. Since $t'$ is a type of $\text{front}(w)$ for which $\alpha(X, t')$ is minimal, we have $\alpha(X, t') \leq \alpha(X, t'') \leq \beta(X, t)$.

Finally, let $v$ be an arbitrary vertex on the path $P$. By the discussion above, we can conclude that for every pair $t_i, t_j \in \text{active}(v)$ the intervals $[\alpha(X, t_i), \beta(X, t_i)]$ and $[\alpha(X, t_j), \beta(X, t_j)]$ overlap, because $t_i$ first appears on path $P$ or $t_j$ first appears on path $P$. Since the cut of all intervals $[\alpha(X, t_i), \beta(X, t_i)], t_i \in \text{active}(v)$ mutually
overlap, there is a bag $X_j$ in the directed path-decomposition $\mathcal{X}$ such that $active(v) \subseteq X_j$.

Example 13 Let $Q = \{q_1, q_2\}$ be the set of sequences $q_1 = [a, b, c, b, d, d, a]$ and $q_2 = [e, f, f, e, c]$ from Example 11 and let $\mathcal{X} = \{(a, e, f), (a, b, c), (d)\}$ be a directed path-decomposition. Then we get the following values for $\alpha(\mathcal{X}, t)$ and $\beta(\mathcal{X}, t)$.

| type $t$ | $a$ | $b$ | $c$ | $d$ | $e$ | $f$ |
|----------|-----|-----|-----|-----|-----|-----|
| $\alpha(\mathcal{X}, t)$ | 1   | 2   | 2   | 3   | 1   | 1   |
| $\beta(\mathcal{X}, t)$ | 2   | 2   | 2   | 3   | 1   | 1   |

The defined path in state digraph $s(Q)$ is shown in Table 4.

By Lemmas 26 and Lemma 27 we obtain the following result.

Corollary 9 Given some set $Q$ of $k$ sequences, then

$$d-pw(g(Q)) = \min_{(v_0, \ldots, v_r) \in \mathcal{P}(Q)} \max_{1 \leq i \leq r-1} |active(v_i)| - 1.$$  

In order to apply Corollary 9 we consider some general digraph problem. Let $G = (V, A, f)$ be a directed acyclic vertex-labeled graph. Function $f : V \rightarrow \mathbb{Z}$ assigns to every vertex $v \in V$ a value $f(v)$. Let $s \in V$ and $t \in V$ be two vertices. For some vertex $v \in V$ and some path $P = (v_1, \ldots, v_\ell)$ with $v_1 = s$, $v_\ell = v$ and $(v_i, v_{i+1}) \in A$ we define $val_P(v) := \max_{u \in P} (f(u))$. Let $\mathcal{P}_s(v)$ denote the set of all

Table 4 Construction of a path $P = (v_0, \ldots, v_r)$ for a given set $Q$ of sequences and a directed path-decomposition of $g(Q)$ of Example 13 to visualize the idea of the proof of Lemma 27

| $j$ | $q_1$ | $q_2$ | $v_j$ | $front(v_j)$ | $active(v_j)$ | $f(v_j) = |active(v_j)|$ |
|-----|-------|-------|-------|--------------|---------------|---------------------|
| 0   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (0,0) | [a,e] | $\emptyset$ | 0 |
| 1   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (1,0) | [b,e] | [a] | 1 |
| 2   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (1,1) | [b,f] | [a,e] | 2 |
| 3   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (1,2) | [b,f] | [a,e,f] | 3 |
| 4   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (1,3) | [b,e] | [a,e] | 2 |
| 5   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (1,4) | [b,c] | [a] | 1 |
| 6   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (1,5) | [b] | [a,c] | 2 |
| 7   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (2,5) | [c] | [a,b,c] | 3 |
| 8   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (3,5) | [b] | [a,b] | 2 |
| 9   | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (4,5) | [d] | [a] | 1 |
| 10  | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (5,5) | [d] | [a,d] | 2 |
| 11  | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (6,5) | [a] | [a] | 1 |
| 12  | [a,b,c,b,d,d,a] | [e,f,f,e,c] | (7,5) | $\emptyset$ | $\emptyset$ | 0 |
paths from vertex \(s\) to vertex \(v\). We define \(\text{val}(v) := \min_{P \in \mathcal{P}_s(v)}(\text{val}_P(v))\). Then it holds:
\[
\text{val}(v) = \max\{f(v), \min_{u \in N^-(v)}(\text{val}(u))\}.
\]

In Fig. 6 we give a dynamic programming solution to compute all the values of \(\text{val}(v), v \in V\) in time \(O(|V| + |A|)\). This is possible, since the graph is directed and acyclic.

Corollary 9 allows us to apply the algorithm given in Fig. 6 on \(s(Q)\) to compute the directed path-width of \(g(Q)\).

**Theorem 5** Given some set \(Q\), such that \(g(Q) \in S_{k,\ell}\) for some \(\ell \geq 1\), then the directed path-width of \(g(Q)\) and a directed path-decomposition can be computed in time \(O(k \cdot (1 + \max_{1 \leq i \leq k} n_i)^k)\).

**Proof** Let \(Q\) be some set, such that \(g(Q) \in S_{k,\ell}\). The state digraph \(s(Q)\) has at most \((1 + \max_{1 \leq i \leq k} n_i)^k\) vertices and can be found in time \(O(k \cdot (1 + \max_{1 \leq i \leq k} n_i)^k)\) from \(Q\). By Corollary 9 the directed path-width of \(g(Q)\) can be computed by considering all paths from the initial state to the final state in \(s(Q)\). This can be done by the algorithm given in Fig. 6 on \(s(Q) = (V, A)\) using \(f(v) = |\text{active}(v)|\), \(v \in V\), \(s\) as the initial state, and \(t\) as the final state. Since every vertex of the state digraph has at most \(k\) outgoing arcs we have \(O(|V| + |A|) \leq O(k \cdot (1 + \max_{1 \leq i \leq k} n_i)^k)\). Thus we can compute an optimal path in \(s(Q)\) in time \(O(k \cdot (1 + \max_{1 \leq i \leq k} n_i)^k)\). □

These results lead to an XP-algorithm for directed path-width w.r.t. number of sequences \(k\) needed to define the input graph. This implies that for each constant \(k\), it is decidable in polynomial time whether for a given set \(Q\) on at most \(k\) sequences the digraph \(g(Q)\) has directed path-width at most \(w\). If we know that some graph can be defined by one sequence, we can find this in linear time (Proposition 6). This implies that for each constant \(k\), it is decidable in polynomial time whether for a digraph

**Algorithm** **Dynamic Programming (DP)**

\[
\text{val}(s) := f(s) \\
\text{for every vertex } v \neq s \text{ in order of topol do } \{ \\
\text{val}[v] := \infty \\
\text{for every } u \in N^-(v) \text{ do } \\
\text{if } (\text{val}[u] < \text{val}[v]) \text{ then } \text{val}[v] := \text{val}[u]; \\
\text{if } (\text{val}[v] < f(v)) \text{ then } \text{val}[v] := f(v) \\
\}
\]

**Fig. 6** Computing a path where the maximum value is as small as possible
\[ G, \text{ which is given by the union of at most } k \text{ many } \{ \text{co-}(2\vec{P}_2), 2\vec{K}_1, \vec{C}_3, D_4 \}\text{-free digraphs, digraph } G \text{ has directed path-width at most } w. \]

### 6 Conclusions and Outlook

In this paper, we have considered digraphs which can be defined by a set of \( k \) sequences. We have shown an XP-algorithm for directed path-width w.r.t. number of sequences \( k \) needed to define the input graph. For special digraphs our solution improves known solution w.r.t. the standard parameter as shown in Table 1. This implies that for each constant \( k \), it is decidable in polynomial time whether for a given set \( Q \) on at most \( k \) sequences the digraph \( g(Q) \) has directed path-width at most \( w \). If we know that some digraph can be defined by one sequence, we can find this in linear time (Proposition 6). This implies that for each constant \( k \), it is decidable in polynomial time whether for a digraph \( G \), which is given by the union of at most \( k \) many semicomplete \( \{ \text{co-}(2\vec{P}_2), \vec{C}_3, D_4 \}\)-free digraphs, digraph \( G \) has directed path-width at most \( w \).

There are several interesting open questions. (a) In Section 5.4 we have shown XP-algorithms for the directed path-width problem w.r.t. parameter \( k \) (number of sequences). It remains open whether there are also FPT-algorithms for this parameter. The same question is open for the standard parameter. (b) Does the hardness of Proposition 11 also hold for \( c_Q \in \{ 2, 3, 4 \} \) and for \( d_Q = 2 \)? (c) By Theorem 1 one can decide in polynomial time whether a given digraph belongs to the class \( S_{1,1} \). By Lemma 12 this is also possible for every class \( S_{k,1} \). Furthermore Theorem 3 allows to recognize the classes \( S_{1,2} \) in polynomial time. It remains to consider this problem for the classes \( S_{k,\ell} \) for \( k \geq 2 \) and \( 1 < \ell \leq 2k \). (d) In order to apply Corollary 9 for some given digraph \( G \) we need a set \( Q \) such that \( g(Q) = G \). Can we find a set \( Q \) with a smallest number of sequences in polynomial time? (e) Can we find a simple set \( Q \) with a smallest number of sequences in polynomial time? (f) By its equivalence to directed vertex separation number \([22]\) directed path-width is a vertex ordering problem (layout problem). This motivates to solve similar problems such as directed cut-width or directed feedback arc set (see \([23]\) for a survey) on sequence digraphs. Is there even a general approach to solve vertex ordering problem on sequence digraphs (see \([23]\) for a general approach on undirected graphs)?

**Acknowledgements** This work was funded in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 388221852

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
References

1. Gurski, F., Rehs, C., Rethmann, J.: Directed pathwidth of sequence digraphs. In: Proceedings of the international conference on combinatorial optimization and applications (COCOA), vol. 11346 of LNCS, Springer, 2018, pp. 79–93 (2018)
2. Kitaev, S., Pyatkin, A.: Word-representable graphs: a survey. J. Appl. Industr. Math. 12(2), 278–296 (2018)
3. Kitsunai, K., Kobayashi, Y., Komuro, K., Tamaki, H., Tano, T.: Computing directed pathwidth in \(O(1.89^n)\) time. Algorithmica 75(2016), 138–157 (2016)
4. Nagamochi, H.: Linear layouts in submodular systems. In: Proceedings of the international symposium on algorithms and computation, vol. 7676 of LNCS, Springer, 2012, pp. 475–484 (2012)
5. Kitsunai, K., Kobayashi, Y., Tamaki, H.: On the pathwidth of almost semicomplete digraphs. In: Proceedings of the annual european symposium on algorithms (ESA), vol. 9294 of LNCS, Springer, 2015, pp. 816–827 (2015)
6. Kobayashi, Y.: Computing the pathwidth of directed graphs with small vertex cover. Inf. Process. Lett. 115(2), 310–312 (2015)
7. Bang-Jensen, J., Gutin, G.: Digraphs. Theory, Algorithms and Applications. Springer, Berlin (2009)
8. Gurski, F., Rethmann, J., Wanke, E.: On the complexity of the FIFO stack-up problem. Math. Methods Oper. Res. 83(1), 33–52 (2016)
9. Gould, R.: Graph Theory. Dover Publications Inc, New York (2012)
10. Harary, F.: Graph Theory. Addison-Wesley Publishing Company, Massachusetts (1969)
11. Barát, J.: Directed pathwidth and monotonicity in digraph searching. Graphs Combinatorics 22(2006), 161–172 (2006)
12. Johnson, T., Robertson, N., Seymour, P., Thomas, R.: Directed tree-width. J. Combinatorial Theory Series B 82(2001), 138–155 (2001)
13. Robertson, N., Seymour, P., Graph Minors, I.: Excluding a forest. J. Combinatorial Theory Series B 35(1983), 39–61 (1983)
14. Kashiybara, T., Fujisawa, T.: NP-completeness of the problem of finding a minimum-clique-number interval graph containing a given graph as a subgraph. In: Proceedings of the international symposium on circuits and systems, 1988, pp. 657–660 (1989)
15. Arnborg, S., Corneil, D., Proskurowski, A.: Complexity of finding embeddings in a k-tree. SIAM J. Algeb. Discrete Methods 8(2), 277–284 (1987)
16. Gusted, J.: On the pathwidth of chordal graphs. Discrete Appl. Math. 45(3), 233–248 (1993)
17. Kloks, T., Bodlaender, H., Müller, H., Kratsch, D.: Computing treewidth and minimum fill-in: all you need are the minimal separators. In: Proceedings of the annual european symposium on algorithms (ESA), vol. 9294 of LNCS, Springer, 2015, pp. 260–271 (1993)
18. Monien, B., Sudborough, I.: Min cut is NP-complete for edge weighted trees. Theor. Comput. Sci. 58(1988), 209–229 (1988)
19. Bodlaender, H.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. 25(6), 1305–1317 (1996)
20. Chudnovsky, M., Fradkin, A., Seymour, P.: Tournament immersion and cutwidth. J. Combinatorial Theory Series B 102(1), 93–101 (2012)
21. Bodlaender, H.: A partial k-arboretum of graphs with bounded treewidth. Theor. Comput. Sci. 209(1998), 1–45 (1998)
22. Yang, B., Cao, Y.: Digraph searching, directed vertex separation and directed pathwidth. Discrete Appl. Math. 156(10), 1822–1837 (2008)
23. Bodlaender, H., Fomin, F., Koster, A., Kratsch, D., Thilikos, D.: A note on exact algorithms for vertex ordering problems on graphs. Theory Comput. Syst. 50(3), 420–432 (2012)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.