2. Classes of Digraphs

In this chapter we introduce several classes of digraphs. We study these classes with respect to their properties, characterization, recognition and decomposition. Further properties of the classes are studied in the following chapters of this book.

In Section 2.1 we study basic properties of acyclic digraphs. Acyclic digraphs form a very important family of digraphs and the reader will often encounter them in this book. Multipartite digraphs and extended digraphs are introduced in Section 2.2. These digraphs are studied in many other sections of our book. In Section 2.3, we introduce and study the transitive closure and a transitive reduction of a digraph. We use the notion of transitive reduction already in Section 2.6.

Several characterizations and a recognition algorithm for line digraphs are given in Section 2.4. We investigate basic properties of de Bruijn and Kautz digraphs and their generalizations in Section 2.5. These digraphs are extreme or almost extreme with respect to their diameter and vertex-strong connectivity. Series-parallel digraphs are introduced and studied in Section 2.6. These digraphs are of interest due to various applications such as scheduling. In the study of series-parallel digraphs we use notions and results of Sections 2.3 and 2.4.

An interesting generalization of transitive digraphs, the class of quasi-transitive digraphs, is considered in Section 2.7. The path-merging property of digraphs which is quite important for investigation of some classes of digraphs including tournaments is introduced and studied in Section 2.8. Two classes of path-mergeable digraphs, locally in-semicomplete and locally out-semicomplete digraphs, both generalizing the class of tournaments, are defined and investigated with respect to their basic properties in Section 2.9. The intersection of these two classes forms the class of locally semicomplete digraphs, which are studied in Section 2.10. There we give a very useful classification of locally semicomplete digraphs, which is applied in several proofs in other chapters. A characterization of a special subclass of locally semicomplete digraphs, called round digraphs, is also proved.

In Section 2.11, we study three classes of totally decomposable digraphs forming important generalizations of quasi-transitive digraphs as well as some other classes of digraphs. The aim of Section 2.11 is to investigate recognition

J. Bang-Jensen, G.Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer Monographs in Mathematics, DOI 10.1007/978-1-84800-998-1_2,

of these three classes. Planar digraphs are discussed in Section 2.12. Digraphs of restricted tree-width are considered in Section 2.13. We show the usefulness of this class of graphs in designing polynomial algorithms and proving fixed-parameter tractability for some problems on digraphs. In Section 2.13, we also introduce and study directed tree-width, directed path-width and DAG-width. The last section is devoted to digraphs of three classes: circulant digraphs, arc-locally semicomplete digraphs and intersection digraphs.

2.1 Acyclic Digraphs

A digraph D is **acyclic** if it has no cycle. Acyclic digraphs form a well-studied family of digraphs of great interest in graph theory, algorithms and applications (see, e.g., Sections 2.3, 2.6, 3.3.2, 7.3, 10.4, 10.7, 17.2, 17.11, 17.15).

Recall that a vertex x in a digraph is sink (source) if $d^+(x) = 0$ ($d^-(x) = 0$).

Proposition 2.1.1 *Every acyclic digraph has a source and a sink.*

Proof: Let D be a digraph in which all vertices have positive out-degrees. We show that D has a cycle. Choose a vertex v_1 in D. Since $d^+(v_1) > 0$, there is a vertex v_2 such that $v_1 \rightarrow v_2$. As $d^+(v_2) > 0$, v_2 dominates some vertex v_3. Proceeding in this manner, we obtain walks of the form $v_1 v_2 \dots v_k$. As $V(D)$ is finite, there exists the least $k > 2$ such that $v_k = v_i$ for some $1 \leq i < k$. Clearly, $v_i v_{i+1} \dots v_k$ is a cycle.

Thus, an acyclic digraph D has a sink. Since the converse H of D is also acyclic, H has a sink v. Clearly, v has a source in D. □

Proposition 2.1.1 allows one to check whether a digraph D is acyclic: if D has a vertex of out-degree zero, then delete this vertex from D and consider the resulting digraph; otherwise, D contains a cycle. In the end of this section, we give another algorithm for verifying whether a digraph is acyclic.

Proposition 2.1.2 *Let D be an acyclic digraph with precisely one source x and one sink y in D. Then for every vertex $v \in V(D)$ there is an (x, v)-path and a (v, y)-path in D.*

Proof: A longest path starting at v (terminating at v) is certainly a (v, y)-path (an (x, v)-path). □

Let D be a digraph and let $x_1, x_2, \dots, x_n$ be an ordering of its vertices. We call this ordering an **acyclic ordering**[1] if, for every arc $x_i x_j$ in D, we

[1] Notice that in a majority of the literature an acyclic ordering is called a topological sorting. We feel that the name acyclic ordering is more appropriate, since no topology is involved.

have $i < j$. Clearly, an acyclic ordering of D induces an acyclic ordering of every subdigraph H of D. Since no cycle has an acyclic ordering, no digraph with a cycle has an acyclic ordering. On the other hand, the following holds:

Proposition 2.1.3 *Every acyclic digraph has an acyclic ordering of its vertices.*

Proof: We give a constructive proof by describing a procedure that generates an acyclic ordering of the vertices in an acyclic digraph D. At the first step, we choose a vertex v with in-degree zero. (Such a vertex exists by Proposition 2.1.1.) Set $x_1 = v$ and delete x_1 from D. At the ith step, we find a vertex u of in-degree zero in the remaining acyclic digraph, set $x_i = u$ and delete x_i from the remaining acyclic digraph. The procedure has $|V(D)|$ steps.

Suppose that $x_i \rightarrow x_j$ in D, but $i > j$. As x_j was chosen before x_i, it means that x_j was not of in-degree zero at the jth step of the procedure; a contradiction. □

Knuth [602] was the first to give a linear time algorithm for finding an acyclic ordering. Now we will show how to find an acyclic ordering in linear time using DFS described in the previous chapter. Below we assume that the input to the DFS algorithm is an acyclic digraph $D = (V, A)$. In the formal description of DFS let us add the following: $i := n + 1$ in line 2 of the main body of DFS and $i := i - 1$, $v_i := v$ in the last line of DFS-PROC. We obtain the following algorithm which we denote by **DFSA**:

DFSA(D)
Input: A digraph $D = (V, A)$.
Output: An acyclic ordering $v_1, \ldots, v_n$ of D.

1. For each $v \in V$ set pred$(v) :=$ **nil**, tvisit$(v) := 0$ and texpl$(v) := 0$.
2. Set $time := 0$, $i := n + 1$.
3. For each vertex $v \in V$ do: if tvisit$(v) = 0$ then perform DFSA-PROC(v).

DFSA-PROC(v)

1. Set $time := time + 1$, tvisit$(v) := time$.
2. For each $u \in N^+(v)$ do: if tvisit$(u) = 0$ then pred$(u) := v$ and perform DFSA-PROC(u).
3. Set $time := time + 1$, texpl$(v) := time$, $i := i - 1$, $v_i := v$.

Theorem 2.1.4 *The algorithm DFSA correctly determines an acyclic ordering of any acyclic digraph in time $O(n + m)$.*

Proof: Since the algorithm is clearly linear (as DFS is linear), it suffices to show that the ordering $v_1, v_2, \ldots, v_n$ is acyclic. Observe that according to our algorithm

$$\text{texpl}(v_i) > \text{texpl}(v_j) \text{ if and only if } i < j. \tag{2.1}$$

Assume that D has an arc v_kv_s such that $k > s$. Hence, $\mathrm{texpl}(v_s) > \mathrm{texpl}(v_k)$. The arc v_kv_s is not a cross arc, because if it were, then by Proposition 1.9.1 and Corollary 1.9.2, the intervals for v_k and v_s would not intersect, i.e., v_k would be visited and explored before v_s would be visited; this and (2.1) make the existence of v_kv_s impossible. The arc v_kv_s is not a forward arc, because if it were, $\mathrm{texpl}(v_s)$ would be smaller than $\mathrm{texpl}(v_k)$. Therefore, v_kv_s must be a backward arc, i.e., $v_k \succ v_s$. Thus, there is a (v_s, v_k)-path in D. This path and the arc v_kv_s form a cycle, a contradiction. □

Figure 2.1 illustrates the result of applying DFSA to an acyclic digraph. The resulting acyclic ordering is z, w, u, y, x, v.

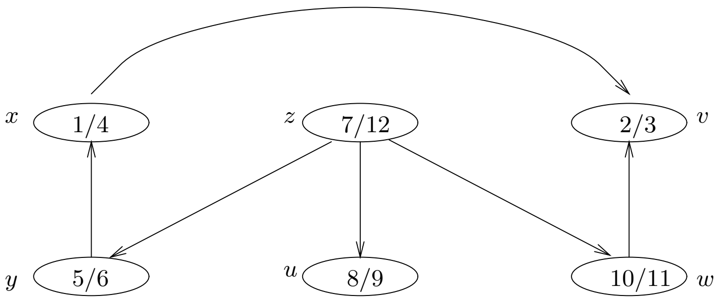


Figure 2.1 The result of applying DFSA to an acyclic digraph.

In Section 5.2 we apply DFSA to an arbitrary not necessarily acyclic digraph and see that the ordering $v_1, v_2, \ldots, v_n$ obtained by DFSA is very useful to determine the strong components of a digraph. DFSA allows us to check, in time $O(n+m)$, whether a digraph D is acyclic: we run DFSA and then verify whether the obtained ordering of the vertices is acyclic. Thus, we have the following:

Proposition 2.1.5 *One can check whether a digraph is acyclic in time $O(n+m)$.* □

2.2 Multipartite Digraphs and Extended Digraphs

A ***p*-partite digraph** is a biorientation of a p-partite graph; see Figure 2.2(b). Bipartite (i.e., 2-partite) digraphs are of special interest. It is well-known (and was proved already by König [618]) that an undirected graph is bipartite if and only if it has no cycle of odd length. The obvious extension of this statement to cycles in digraphs is not valid (the non-bipartite digraph with vertex set $\{x, y, z\}$ and arc set $\{xy, xz, yz\}$ is such an example that can easily be generalized). However, the obvious extension does hold for strong bipartite digraphs. Theorem 2.2.1 can be traced back to the book [503] by Harary, Norman and Cartwright.

Theorem 2.2.1 *A strongly connected digraph is bipartite if and only if it has no cycle of odd length.*

Proof: If D is bipartite, then it is easy to see that D cannot have an odd cycle. To prove sufficiency suppose that D has no odd cycle. Fix an arbitrary vertex x in D. We claim that for every vertex $y \in V(D) - x$ and every choice of an (x, y)-path P and a (y, x)-path Q, the length of P and Q are equal modulo 2.

Suppose this is not the case for some choice of y, P and Q. Then choose y, P and Q such that the parity of the lengths of P and Q differ and $|V(P)| + |V(Q)|$ is minimum among all such pairs of (x, y)- and (y, x)-paths whose lengths differ in parity. If $V(P) \cap V(Q) = \{x, y\}$, then PQ is an odd cycle, contradicting the assumption. Hence there is a vertex $z \notin \{x, y\}$ in $V(P) \cap V(Q)$. Let z be chosen as the first such vertex that we meet when we traverse Q from y towards x. Then $P[z, y]Q[y^+_Q, z]$ is a cycle and it is even by our assumption. But now it is easy to see that the parity of the paths $P[x, z]$ and $Q[z, x]$ are different and we get a contradiction to the choice of y, P and Q. This proves the claim and, in particular, it follows that for every $y \in V(D) - x$, the lengths of all paths from x to y have the same parity.

Now let $U = \{y : \text{ the length of every } (x, y)\text{-path is even}\}$ and $U' = \{y : \text{ the length of every } (x, y)\text{-path is odd}\}$. This is a bipartition of $V(D)$ and neither U nor U' contains two vertices which are joined by an arc, since that would imply that some vertex had paths of two different parities from x. $\square$

An extension of this theorem to digraphs whose cycles are all of length 0 modulo $k \geq 2$ is given in Theorem 17.8.1.

Recall that tournaments are orientations of complete graphs. Recall that a **semicomplete digraph** is a biorientation of a complete graph (see Figure 2.2(a)) and a **tournament** is an orientation of a complete digraph. The complete biorientation of a complete graph is a **complete digraph** (denoted by $\overleftrightarrow{K}_n$). The notion of semicomplete digraphs and their special subclass, tournaments, can be extended in various ways. A biorientation of a complete p-partite (multipartite) graph is a **semicomplete p-partite (multipartite) digraph**; see Figure 2.2(c). A **multipartite tournament** is an orientation of a complete multipartite graph. A semicomplete 2-partite digraph is also called a **semicomplete bipartite digraph**. A **bipartite tournament** is a semicomplete bipartite digraph with no 2-cycles.

One can use the operation of extension introduced in Section 1.3 to define 'extensions' of the above classes of digraphs. We will speak of **extended semicomplete digraphs** (i.e., extensions of semicomplete digraphs), **extended locally in-semicomplete digraphs**, **extended locally semi-**

(a) K_4 and a semicomplete digraph of order four.

(b) A 3-partite graph G and a biorientation of G.

(c) The complete 3-partite graph $K_{2,1,2}$ and
a semicomplete 3-partite digraph D.

Figure 2.2 Multipartite digraphs.

complete digraphs, etc. Clearly, every extended semicomplete digraph is a semicomplete multipartite digraph, which is not necessarily semicomplete, and every extended semicomplete multipartite digraph is still a semicomplete multipartite digraph. Therefore, the class of semicomplete multipartite digraphs is **extension-closed**, and the class of semicomplete digraphs is not.

2.3 Transitive Digraphs, Transitive Closures and Reductions

A digraph D is **transitive** if, for every pair xy and yz of arcs in D with $x \neq z$, the arc xz is also in D. Transitive digraphs form the underlying graph-theoretical model in a number of applications. For example, transitive oriented graphs correspond very naturally to partial orders (see Section 13.5 for some terminology on partial orders, the correspondence between transitive oriented graphs and partial orders and some basic results). The aim of this section is to give a brief overview of some properties of transitive digraphs as well as transitive closures and reductions of digraphs.

It is easy to show that a tournament is transitive if and only it is acyclic (see Exercise 2.3) and a strong digraph D is transitive if and only if D is complete[2]. We have the following simple characterization of transitive digraphs; its proof is left as Exercise 2.4.

Proposition 2.3.1 *Let D be a digraph with an acyclic ordering $D_1, D_2, \dots, D_p$ of its strong components. The digraph D is transitive if and only if each of D_i is complete, the digraph H obtained from D by contraction of $D_1, \dots, D_p$ followed by deletion of multiple arcs is a transitive oriented graph, and $D = H[D_1, D_2, \dots, D_p]$, where $p = |V(H)|$.* □

The **transitive closure** $TC(D)$ of a digraph D is a digraph with $V(TC(D)) = V(D)$ and, for distinct vertices u, v, the arc $uv \in A(TC(D))$ if and only if D has a (u, v)-path. Clearly, if D is strong, then $TC(D)$ is a complete digraph. The uniqueness of the transitive closure of an arbitrary digraph is obvious. To compute the transitive closure of a digraph one can obviously apply the Floyd-Warshall algorithm in Chapter 3. To obtain a faster algorithm for the problem one can use the fact discovered by a number of researchers (see, e.g., the paper [318] by Fisher and Meyer, or [370] by Furman) that the transitive closure problem and the matrix multiplication problem are closely related: there exists an $O(n^a)$-algorithm, with $a \geq 2$, to compute the transitive closure of a digraph of order n if and only if the product of two boolean $n \times n$ matrices can be computed in $O(n^a)$ time. Coppersmith and Winograd [230] showed that there exists an $O(n^{2.376})$-algorithm for the matrix multiplication. Goralcikova and Koubek [423] designed an $O(nm_{red})$-algorithm to find the transitive closure of an acyclic digraph D with n vertices and m_{red} arcs in the transitive reduction of D (the notion of transitive reduction is introduced below). This algorithm was also studied and improved by Mehlhorn [691] and Simon [820].

An arc uv in a digraph D is **redundant** if there is a (u, v)-path in D which does not contain the arc uv. A **transitive reduction** of a digraph D is a spanning subdigraph H of D with no redundant arc such that the transitive closures of D and H coincide. Not every digraph D has a unique transitive reduction. Indeed, if D has a pair of hamiltonian cycles, then each of these cycles is a transitive reduction of D. Below we show that a transitive reduction of an acyclic digraph is unique, i.e., we may speak of *the* transitive

[2] By the definition of a transitive digraph, a 2-cycle xyx does not force a loop at x and y.

reduction of an acyclic digraph. The **intersection of digraphs** $D_1, \dots, D_k$ with the same vertex set V is the digraph H with vertex set V and arc set $A(D_1) \cap \dots \cap A(D_k)$. Similarly one can define the union of digraphs with the same vertex sets (see Section 1.3). Let $\mathcal{S}(D)$ be the set of all spanning subdigraphs L of D for which $TC(L) = TC(D)$.

Theorem 2.3.2 *[10] For an acyclic digraph D, there exists a unique digraph D' with the property that $TC(D') = TC(D)$ and every proper subdigraph H of D' satisfies $TC(H) \neq TC(D')$. The digraph D' is the intersection of digraphs in $\mathcal{S}(D)$.*

The proof of this theorem, which is due to Aho, Garey and Ullman, follows from Lemmas 2.3.3 and 2.3.4.

Lemma 2.3.3 *Let D and H be a pair of acyclic digraphs on the same vertex set such that $TC(D) = TC(H)$ and $A(D) - A(H) \neq \emptyset$. Then, for every $e \in A(D) - A(H)$, we have $TC(D - e) = TC(D)$.*

Proof: Let $e = xy \in A(D) - A(H)$. Since $e \notin A(H)$, H must have an (x, y)-path passing through some other vertex, say z. Hence, D has an (x, z)-path P_{xz} and a (z, y)-path P_{zy}. If P_{xz} contains e, then D has a (y, z)-path. The existence of this path contradicts the existence of P_{zy} and the hypothesis that D is acyclic. Similarly, one can show that P_{zy} does not contain e. Therefore, $D - e$ has an (x, y)-path. Hence, $TC(D - e) = TC(D)$. □

Lemma 2.3.4 *Let D be an acyclic digraph. Then the set $\mathcal{S}(D)$ is closed under union and intersection.*

Proof: Let G, H be a pair of digraphs in $\mathcal{S}(D)$. Since $TC(G) = TC(H) = TC(D)$, $G \cup H$ is a subdigraph of $TC(D)$. The transitivity of $TC(D)$ now implies that $TC(G \cup H)$ is a subdigraph of $TC(D)$. Since G is a subdigraph of $G \cup H$, we have $TC(D)$ $(= TC(G))$ is a subdigraph of $TC(G \cup H)$. Thus, we conclude that $TC(G \cup H) = TC(D)$ and $G \cup H \in \mathcal{S}(D)$.

Now let $e_1, \dots, e_p$ be the arcs of $G - A(G \cap H)$. By repeated application of Lemma 2.3.3, we obtain $TC(G - e_1 - e_2 - \dots - e_p) = TC(G)$. This means that $TC(G \cap H) = TC(G) = TC(D)$, hence $G \cap H \in \mathcal{S}(D)$. □

Aho, Garey and Ullman [10] proved that there exists an $O(n^a)$-algorithm, with $a \geq 2$, to compute the transitive closure of an arbitrary digraph D of order n if and only if a transitive reduction of D can be constructed in time $O(n^a)$. Therefore, we have

Proposition 2.3.5 *For an arbitrary digraph D, the transitive closure and a transitive reduction can be computed in time $O(n^{2.376})$.* □

Simon [821] described an $O(n+m)$-algorithm to find a transitive reduction of a strong digraph D. The algorithm uses DFS and two digraph transformations preserving $TC(D)$. This means that to have a linear time algorithm for finding transitive reductions of digraphs from a certain class $\mathcal{D}$, it suffices to design a linear time algorithm for the transitive reduction of strong component digraphs of digraphs in $\mathcal{D}$. (Recall that the strong component digraph $SC(D)$ of a digraph D is obtained by contracting every strong component of D to a vertex followed by deletion of parallel arcs.) Such algorithms are considered, e.g., in the paper [485] by Habib, Morvan and Rampon.

While Simon's linear time algorithm in [821] finds a *minimal* subdigraph D' of a strong digraph D such that $TC(D') = TC(D)$, no polynomial algorithm is known to find a subdigraph D'' of a strong digraph D with *minimum* number of arcs such that $TC(D'') = TC(D)$. This is not surprising due to the fact that the corresponding optimization problem is $\mathcal{NP}$-hard. Indeed, the problem to verify whether a strong digraph D of order n has a subdigraph D'' of size n such that $TC(D'') = TC(D)$ is equivalent to the hamiltonian cycle problem, which is $\mathcal{NP}$-complete by Theorem 6.1.1.

A subdigraph D'' of a digraph D with minimum number of arcs such that $TC(D'') = TC(D)$ is sometimes called a **minimum equivalent subdigraph** of D. By the above discussion, we see that a minimum equivalent subdigraph of an acyclic digraph is unique and can be found in polynomial time. This means that the main difficulty of finding a minimum equivalent subdigraph of an arbitrary digraph D lies in finding such subdigraphs for the strong components of D. This issue is addressed in Section 12.2 for some classes of digraphs studied in this chapter. For the classes in Section 12.2, the minimum equivalent subdigraph problem is polynomial time solvable.

2.4 Line Digraphs

For a directed pseudograph D, the **line digraph** $Q = L(D)$ has vertex set $V(Q) = A(D)$ and arc set

$$A(Q) = \{ab : a, b \in V(Q), \text{ the head of } a \text{ coincides with the tail of } b\}.$$

A directed pseudograph H is a **line digraph** if there is a directed pseudograph D such that $H = L(D)$. See Figure 2.3. Clearly, line digraphs do not have parallel arcs; moreover, the line digraph $L(D)$ has a loop at a vertex $a \in A(D)$ if and only if a is a loop in D.

The following theorem provides a number of equivalent characterizations of line digraphs. Of these characterizations, (ii) is due to Harary and Norman [502], (iii) to Heuchenne [522] and (iv) and (v) to Richards [777]; conditions (ii) and (iii) have each been rediscovered several times, see the survey [516] by Hemminger and Beineke. The proof presented here is adapted from [516]. For an $n \times n$-matrix $M = [m_{ik}]$, a row i is **orthogonal** to a row j if $\sum_{k=1}^{n} m_{ik}m_{jk} = 0$. One can give a similar definition of orthogonal columns.

Theorem 2.4.1 *Let D be a directed pseudograph with vertex set $\{1, 2, \ldots, n\}$ and with no parallel arcs and let $M = [m_{ij}]$ be its adjacency matrix (i.e., the*

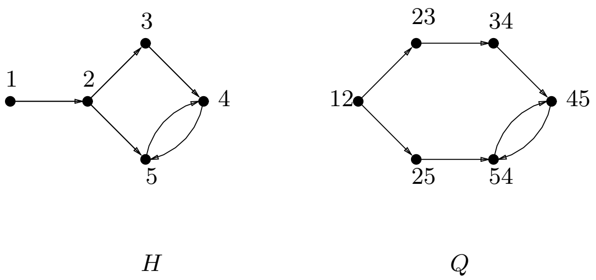


Figure 2.3 A digraph H and its line digraph $Q = L(H)$.

$n \times n$-matrix such that $m_{ij} = 1$, if $ij \in A(D)$, and $m_{ij} = 0$, otherwise). Then the following assertions are equivalent:

(i) D is a line digraph;
(ii) there exist two partitions $\{A_i\}_{i\in I}$ and $\{B_i\}_{i\in I}$ of $V(D)$ such that

$$A(D) = \cup_{i\in I} A_i \times B_i;$$

(iii) if vw, uw and ux are arcs of D, then so is vx;
(iv) any two rows of M are either identical or orthogonal;
(v) any two columns of M are either identical or orthogonal.

Proof: We show the following implications and equivalences: (i) $\Leftrightarrow$ (ii), (ii) $\Rightarrow$ (iii), (iii) $\Rightarrow$ (iv), (iv) $\Leftrightarrow$ (v), (iv) $\Rightarrow$ (ii).

(i) $\Rightarrow$ (ii). Let $D = L(H)$. For each $v_i \in V(H)$, let A_i and B_i be the sets of in-coming and out-going arcs at v_i, respectively. Then the arc set of the subdigraph of D induced by $A_i \cup B_i$ equals $A_i \times B_i$. If $ab \in A(D)$, then there is an i such that $a = v_jv_i$ and $b = v_iv_k$. Hence, $ab \in A_i \times B_i$. The result follows.

(ii) $\Rightarrow$ (i). Let Q be the directed pseudograph with ordered pairs (A_i, B_i) as vertices, and with $|A_j \cap B_i|$ arcs from (A_i, B_i) to (A_j, B_j) for each i and j (including $i = j$). Let σ_{ij} be a bijection from $A_j \cap B_i$ to this set of arcs (from (A_i, B_i) to (A_j, B_j)) of Q. Then the function σ defined on $V(D)$ by taking σ to be σ_{ij} on $A_j \cap B_i$ is a well-defined function of $V(D)$ into $V(L(Q))$, since $\{A_j \cap B_i\}_{i,j\in I}$ is a partition of $V(D)$. Moreover, σ is a bijection since every σ_{ij} is a bijection. Furthermore, it is not difficult to see that σ is an isomorphism from D to $L(Q)$ (this is left as Exercise 2.6).

(ii) $\Rightarrow$ (iii). If vw, uw and ux are arcs of D, then there exist i, j such that $\{u, v\} \subseteq A_i$ and $\{w, x\} \subseteq B_j$. Hence, $(v, x) \in A_i \times B_j$ and $vx \in D$.

(iii) $\Rightarrow$ (iv). Assume that (iv) does not hold. This means that some rows, say i and j, are neither identical nor orthogonal. Then there exist k, h such that $m_{ik} = m_{jk} = 1$ and $m_{ih} = 1, m_{jh} = 0$ (or vice versa). Hence, ik, jk, ih are in $A(D)$ but jh is not. This contradicts (iii).

(iv) $\Leftrightarrow$ (v). Both (iv) and (v) are equivalent to the statement:

$$\text{for all } i, j, h, k, \text{ if } m_{ih} = m_{ik} = m_{jk} = 1, \text{ then } m_{jh} = 1.$$

(iv) $\Rightarrow$ (ii). For each i and j with $m_{ij} = 1$, let $A_{ij} = \{h : m_{hj} = 1\}$ and $B_{ij} = \{k : m_{ik} = 1\}$. Then, by (iv), A_{ij} is the set of vertices in D whose row vectors in M are identical to the ith row vector, whereas B_{ij} is the set of vertices in D whose column vectors in M are identical to the jth column vector (we use the previously proved fact that (iv) and (v) are equivalent). Thus, $A_{ij} \times B_{ij} \subseteq A(D)$, and moreover $A(D) = \cup\{A_{ij} \times B_{ij} : m_{ij} = 1\}$. By the orthogonality condition, A_{ij} and A_{hk} are either equal or disjoint, as are B_{ij} and B_{hk}. For zero row vector i in M, let A_{ij} be the set of vertices whose row vector in M is the zero vector, and let $B_{ij} = \emptyset$. Doing the same with the zero column vectors of M completes the partition as in (ii). □

The characterizations (ii)-(v) all imply polynomial algorithms to verify whether a given directed pseudograph is a line digraph. This fact is obvious regarding (iii)-(v); it is slightly more difficult to see that (ii) can be used to construct a very effective polynomial algorithm. We actually design such an algorithm for acyclic digraphs (as a pair of procedures illustrated by an example) just after Proposition 2.4.3. The criterion (iii) also provides the following characterization of line digraphs in terms of forbidden induced subdigraphs. Its proof is left as Exercise 2.7.

Corollary 2.4.2 *A directed pseudograph D is a line digraph if and only if D does not contain, as an induced subdigraph, any directed pseudograph that can be obtained from one of the directed pseudographs in Figure 2.4 (dotted arcs are missing) by adding zero or more arcs (other than the dotted ones).* □

Observe that the digraph of order 4 in Figure 2.4 corresponds to the case of distinct vertices in Part (iii) of Theorem 2.4.1, and the two directed pseudographs of order 2 correspond to the cases $x = u \neq v = w$ and $u = w \neq v = x$, respectively.

Clearly, Theorem 2.4.1 implies a set of characterizations of the line digraphs of digraphs (without parallel arcs and loops). This can be found in [516]. Several characterizations of special classes of line digraphs and iterated line digraphs can be found in surveys by Hemminger and Beineke [516] and Prisner [755].

Many applications of line digraphs deal with the line digraphs of special families of digraphs, for example regular digraphs, in general, and complete digraphs, in particular, see, e.g., the papers [279] by Du, Lyuu and Hsu and [316] by Fiol, Yebra and Alegre. In Section 2.6, we need the following characterization, due to Harary and Norman, of the line digraphs of acyclic directed multigraphs. It is a specialization of Parts (i) and (ii) of Theorem 2.4.1. The proof is left as (an easy) Exercise 2.8.

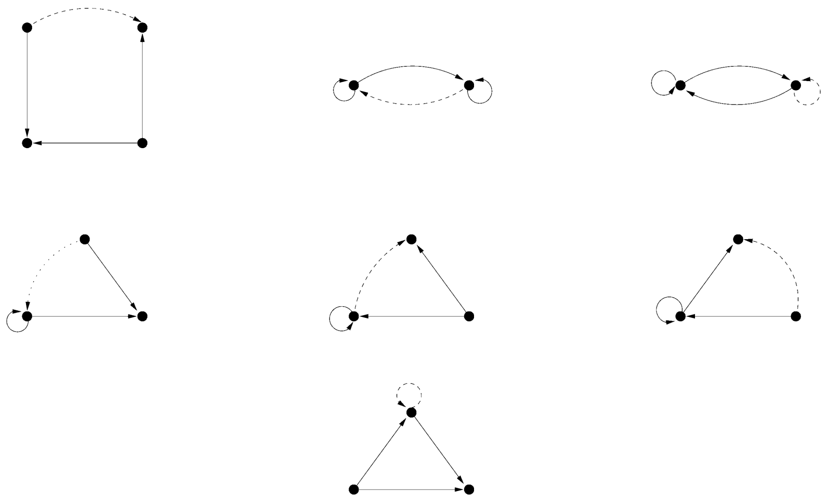


Figure 2.4 Forbidden directed pseudographs.

Proposition 2.4.3 *[502] A digraph D is the line digraph of an acyclic directed multigraph if and only if D is acyclic and there exist two partitions $\{A_i\}_{i\in I}$ and $\{B_i\}_{i\in I}$ of $V(D)$ such that $A(D) = \cup_{i\in I} A_i \times B_i$.* □

We will now show how Proposition 2.4.3 can be used to recognize very effectively whether a given acyclic digraph R is the line digraph of another acyclic directed multigraph H, i.e., $R = L(H)$. The two procedures, which we construct and illustrate by Figure 2.7, can actually be used to recognize and represent (that is, to construct H such that $R = L(H)$) arbitrary line digraphs (see Theorem 2.4.1(i) and (ii)).

We first use Proposition 2.4.3 to check whether H above exists. The following procedure **Check-H** can be applied. Initially, all arcs and vertices of R are not marked. At every iteration, we choose an arc uv in R, which is not marked yet, and mark all vertices in $N^+(u)$ by 'B', all vertices in $N^-(v)$ by 'A' and all arcs in $(N^-(v), N^+(u))_R$ by 'C'. If $(N^-(v), N^+(u))_R \neq N^-(v) \times N^+(u)$ or if we mark a certain vertex or arc twice (starting from another arc $u'v'$) by the same symbol, then this procedure stops as there is no H such that $L(H) = R$. (We call these conditions **obstructions**.) If this procedure is performed to the end (i.e., every vertex and arc received a mark), then such H exists. It is not difficult to see, using Proposition 2.4.3, that Check-H correctly verifies whether H exists or not.

To illustrate Check-H, consider the digraph R_0 of Figure 2.7(a). Suppose that we choose the arc ab first. Then ab is marked, at the first iteration, together with the arcs af and ag. The vertex a receives 'A', the vertices b, f, g get 'B'. Suppose that fi is chosen at the second iteration. Then the arcs fh, fi, gh, gi are all marked at this iteration. The vertices f, g receive

'A', the vertices h, i 'B'. Suppose that bc is chosen at the third iteration. We see that this arc is the only arc marked at this iteration. The vertex b receives 'A', the vertex c 'B'. Finally, say, ce is chosen. Then both cd and ce are marked. The vertex c gets 'A', the vertices d, e receive 'B'. Thus, all arcs have been marked and no obstruction has taken place. This means that there exists a digraph H_0 such that $H_0 = L(R_0)$.

Suppose now that H does exist. The following procedure **Build-H** constructs such a directed multigraph H. By Proposition 2.4.3, if H exists, then all arcs of R can be partitioned into arc sets of bipartite tournaments with partite sets A_i and B_i and arc sets $A_i \times B_i$. Let us denote these digraphs by $T_1, \ldots, T_k$. (They can be computed by Check-H if we mark every $(N^-(v), N^+(u))_R$ not only by 'C' but also by a second mark 'i' starting from 1 and increasing by 1 at each iteration of the procedure.) We construct H as follows. The vertex set of H is $\{t_0, t_1, \ldots, t_k, t_{k+1}\}$. The arcs of H are obtained by the following procedure. For each vertex v of R, we append one arc a_v to H according to the rules below:

(a) If $d_R(v) = 0$, then $a_v := (t_0, t_{k+1})$;
(b) If $d^+_R(v) > 0, d^-_R(v) = 0$, then $a_v := (t_0, t_i)$, where i is the index of T_i such that $v \in A_i$;
(c) If $d^+_R(v) = 0, d^-_R(v) > 0$, then $a_v := (t_j, t_{k+1})$, where j is the index of T_j such that $v \in B_j$;
(d) If $d^+_R(v) > 0, d^-_R(v) > 0$, then $a_v := (t_i, t_j)$, where i and j are the indices of T_i and T_j such that $v \in A_j \cap B_i$.

It is straightforward to verify that $R = L(H)$. Note that Build-H always constructs H with only one vertex of in-degree zero and only one vertex of out-degree zero.

To illustrate Build-H, consider R_0 of Figure 2.7 once again. Earlier we showed that there exists H_0 such that $R_0 = L(H_0)$. Now we will construct H_0. The previous procedure applied to verify the existence of H_0 has implicitly constructed the digraphs $T_1 = (\{a, b, f, g\}, \{ab, af, ag\})$, $T_2 = (\{f, g, h, i\}, \{fh, fi, gh, gi\})$, $T_3 = (\{b, c\}, \{bc\})$, $T_4 = (\{c, d, e\}, \{cd, ce\})$. Thus, H_0 has vertices $t_0, \ldots, t_5$. Considering the vertices of R_0 in the lexicographic order, we obtain the following arcs of H_0 (in this order):

$$t_0t_1, t_1t_3, t_3t_4, t_4t_5, t_4t_5, t_1t_2, t_1t_2, t_2t_5, t_2t_5.$$

The directed multigraph H_0 is depicted in Figure 2.7(c). It is easy to check that $R_0 = L(H_0)$.

The iterated line digraphs are defined recursively: $L^1(D) = L(D)$, $L^{k+1}(D) = L(L^k(D))$, $k \geq 1$. It is not difficult to prove by induction (Exercise 2.10) that $L^k(D)$ is isomorphic to the digraph H, whose vertex set consists of walks of D of length k and a vertex $v_0v_1 \ldots v_k$ (which is a walk in D) dominates the vertex $v_1v_2 \ldots v_kv_{k+1}$ for every $v_{k+1} \in V(D)$ such that

$v_k v_{k+1} \in A(D)$. New characterizations of line digraphs and iterated line digraphs are given by Liu and West [648].

The following proposition can be proved by induction on $k \geq 1$ (Exercise 2.12).

Proposition 2.4.4 *Let D be a strong d-regular digraph ($d > 1$) of order n and diameter t. Then $L^k(D)$ is of order $d^k n$ and diameter $t + k$.* □

Bermond, Munos and Marchetti-Spaccamela [150] proposed broadcasting algorithms for line digraphs in the telephone mode. The protocols of [150] use a broadcasting protocol for a digraph D to obtain a broadcasting protocol for iterated line digraphs of D. As a consequence, improved bounds for the broadcasting time in de Bruijn and Kautz digraphs were obtained.

2.5 The de Bruijn and Kautz Digraphs

The following problem is of importance in network design. Given positive integers n and d, construct a digraph D of order n and maximum out-degree at most d such that $\mathrm{diam}(D)$ is as small as possible and the vertex-strong connectivity $\kappa(D)$ is as large as possible. So we have a 2-objective optimization problem. For such a problem, in general, no solution can maximize/minimize both objective functions. However, for this specific problem, there are solutions, which (almost) maximize/minimize both objective functions. The aim of this section is to introduce these solutions, the de Bruijn and Kautz digraphs, as well as some of their generalizations. For more information on the above classes of digraphs, the reader may consult the survey [276] by Du, Cao and Hsu. For applications of these digraphs in design of parallel architectures and large packet radio networks, see e.g. the papers [149] by Bermond and Hell, [151] by Bermond and Peyrat and [792] by Samatan and Pradhan.

Let V be the set of vectors with t coordinates, $t \geq 2$, each taken from $\{0, 1, \ldots d-1\}$, $d \geq 2$. The **de Bruijn digraph** $D_B(d,t)$ is the directed pseudograph with vertex set V such that $(x_1, x_2, \ldots, x_t)$ dominates $(y_1, y_2, \ldots, y_t)$ if and only if $x_2 = y_1, x_3 = y_2, \ldots, x_t = y_{t-1}$. See Figure 2.5(a). Let $D_B(d,1)$ be the complete digraph of order d with loop at every vertex.

These directed pseudographs are named after de Bruijn who was the first to consider them in [252]. Clearly, $D_B(d,t)$ has d^t vertices and the out-pseudodegree and in-pseudodegree of every vertex of $D_B(d,t)$ equal d. This directed pseudograph has no parallel arcs and contains a loop at every vertex for which all coordinates are the same. It is natural to call $D_B(d,t)$ ***d*-pseudoregular** (recall that in the definition of semi-degrees we do not count loops).

Since $D_B(d,t)$ has loops at some vertices, the vertex-strong connectivity of $D_B(d,t)$ is at most $d-1$ (indeed, the loops can be deleted without the vertex-strong connectivity being changed). Imase, Soneoka and Okada [550]

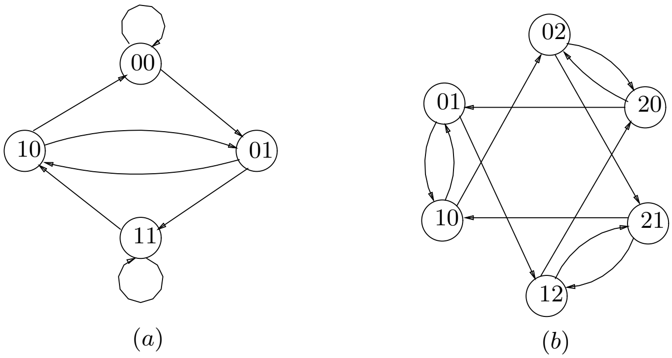


Figure 2.5 (a) The de Bruijn digraph $D_B(2,2)$; (b) the Kautz digraph $D_K(2,2)$.

proved that $D_B(d,t)$ is $(d-1)$-strong, and moreover, for every pair $x \neq y$ of vertices there exist $d-1$ internally disjoint (x,y)-paths of length at most $t+1$. To prove this result we will use the following two lemmas. The proof of the first lemma, due to Fiol, Yebra and Alegre, is left as Exercise 2.13.

Lemma 2.5.1 *[316] For $t \geq 2$, $D_B(d,t)$ is the line digraph of $D_B(d,t-1)$.* □

Lemma 2.5.2 *Let x,y be distinct vertices of $D_B(d,t)$ such that $x \rightarrow y$. Then, there are $d-2$ internally disjoint (x,y)-paths different from xy, each of length at most $t+1$.*

Proof: Let $x = (x_1, x_2, \ldots, x_t)$ and $y = (x_2, \ldots, x_t, y_t)$. Consider the walk W_k given by $W_k = (x_1, x_2, \ldots, x_t), (x_2, \ldots, x_t, k), (x_3, \ldots, x_t, k, x_2), \ldots, (k, x_2, \ldots, x_t), (x_2, \ldots, x_t, y_t)$, where $k \neq x_1, y_t$. For each k, every internal vertex of W_k has coordinates forming the same multiset $M_k = \{x_2, \ldots, x_t, k\}$. Since for different k, the multisets M_k are different, the walks W_k are internally disjoint. Each of these walks is of length $t+1$. Therefore, by Proposition 1.4.1, $D_B(d,t)$ contains $d-2$ internally disjoint (x,y)-paths P_k with $A(P_k) \subseteq A(W_k)$. Since $k \neq x_1, y_t$, we may form the paths P_k such that none of them coincides with xy. □

Theorem 2.5.3 *[550] For every pair x,y of distinct vertices of $D_B(d,t)$, there exist $d-1$ internally disjoint (x,y)-paths, one of length at most t and the others of length at most $t+1$.*

Proof: By induction on $t \geq 1$. Clearly, the claim holds for $t = 1$ since $D_B(d,1)$ contains, as spanning subdigraph, $\overleftrightarrow{K}_d$. For $t \geq 2$, by Lemma 2.5.1, we have that

$$D_B(d,t) = L(D_B(d,t-1)). \tag{2.2}$$

Let x, y be a pair of distinct vertices in $D_B(d,t)$ and let e_x, e_y be the arcs of $D_B(d,t-1)$ corresponding to vertices x, y due to (2.2). Let u be the head of e_x and let v be the tail of e_y.

If $u \neq v$, by the induction hypothesis, $D_B(d,t-1)$ has $d-1$ internally disjoint (u,v)-paths, one of length at most $t-1$ and the others of length at most t. The arcs of these paths together with arcs e_x and e_y correspond to $d-1$ internally disjoint (x,y)-paths in $D_B(d,t)$, one of length at most t and the others of length at most $t+1$.

If $u = v$, we have $x \rightarrow y$ in $D_B(d,t-1)$. It suffices to apply Lemma 2.5.2 to see that there are $d-1$ internally disjoint (x,y)-paths in $D_B(d,t)$, one of length one and the others of length at most $t+1$. □

By this theorem and Corollary 5.4.2, we conclude that $\kappa(D_B(d,t)) = d-1$. From Theorem 2.5.3 and Proposition 3.4.3, we obtain immediately the following simple, yet important property.

Proposition 2.5.4 *The de Bruijn digraph $D_B(d,t)$ achieves the minimum value t of diameter for directed pseudographs of order d^t and maximum out-degree at most d.* □

For $t \geq 2$, the **Kautz digraph** $D_K(d,t)$ is obtained from $D_B(d+1,t)$ by deletion of all vertices of the form $(x_1, x_2, \ldots, x_t)$ such that $x_i = x_{i+1}$ for some i. See Figure 2.5(b). Define $D_K(d,1) := \overleftrightarrow{K}_{d+1}$. Clearly, $D_K(d,t)$ has no loops and is a d-regular digraph. Since we have $d+1$ choices for the first coordinate of a vertex in $D_K(d,t)$ and d choices for each of the other coordinates, the order of $D_K(d,t)$ is $(d+1)d^{t-1} = d^t + d^{t-1}$. It is easy to see that Proposition 2.5.4 holds for the Kautz digraphs as well.

The following lemmas are analogous to Lemmas 2.5.1 and 2.5.2. Their proofs are left as Exercises 2.14 and 2.15.

Lemma 2.5.5 *For $t \geq 2$, the Kautz digraph $D_K(d,t)$ is the line digraph of $D_K(d,t-1)$.* □

Lemma 2.5.6 *Let xy be an arc in $D_K(d,t)$. There are $d-1$ internally disjoint (x,y)-paths different from xy, one of length at most $t+2$ and the others of length at most $t+1$.* □

The following result due to Du, Cao and Hsu [276] shows that the Kautz digraphs are better, in a sense, than de Bruijn digraphs from the local vertex-strong connectivity point of view. This theorem can be proved similarly to Theorem 2.5.3 and is left as Exercise 2.16.

Theorem 2.5.7 *[276] Let x, y be distinct vertices of $D_K(d,t)$. Then there are d internally disjoint (x,y)-paths in $D_K(d,t)$, one of length at most t, one of length at most $t+2$ and the others of length at most $t+1$.* □

This theorem implies that $D_K(d,t)$ is d-strong.

The de Bruijn digraphs were generalized independently by Imase and Itoh [547] and Reddy, Pradhan and Kuhl [767] in the following way. We can transform every vector $(x_1, x_2, \ldots, x_t)$ with coordinates from $Z_d = \{0, 1, \ldots, d-1\}$ into an integer from $Z_{d^t} = \{0, 1, \ldots, d^t - 1\}$ using the polynomial $P(x_1, x_2, \ldots, x_t) = x_1 d^{t-1} + x_2 d^{t-2} + \ldots + x_t$. It is easy to see that this polynomial provides a bijection from Z_d^t to Z_{d^t}. Moreover, for $i, j \in Z_{d^t}$, $i \rightarrow j$ in $D_B(d,t)$ if and only if $j \equiv di + k \pmod{d^t}$ for some $k \in Z_d$.

Let d, n be two natural numbers such that $d < n$. The **generalized de Bruijn digraph** $D_G(d,n)$ is a directed pseudograph with vertex set Z_n and arc set

$$\{(i, di + k \pmod{n}) : \ i, k \in Z_d\}.$$

For example, $V(D_G(2,5)) = \{0,1,2,3,4\}$ and $A(D_G(2,5)) = \{(0,0),(0,1), (1,2),(1,3),(2,4),(2,0),(3,1),(3,2),(4,3),(4,4)\}$.

Clearly, $D_G(d,n)$ is d-pseudoregular. It is not difficult to show that $\mathrm{diam}(D_G(d,n)) \leq \lceil \log_d n \rceil$. By Proposition 3.4.3, a digraph of maximum out-degree at most $d \geq 2$ and order n has a diameter at least $\lfloor \log_d n(d-1) + 1 \rfloor$. Thus, the generalized de Bruijn digraphs are of optimal or almost optimal diameter. It was proved, by Imase, Soneoka and Okada [549], that $D_G(d,n)$ is $(d-1)$-strong. It follows from these results that the generalized de Bruijn digraphs have almost minimum diameter and almost maximum vertex-strong connectivity.

The Kautz digraphs were generalized by Imase and Itoh [548]. Let n, d be two natural numbers such that $d < n$. The Imase-Itoh digraph $D_I(d,n)$ is the digraph with vertex set $\{0, 1, \ldots, n-1\}$ such that $i \rightarrow j$ if and only if $j \equiv -d(i+1) + k \pmod{n}$ for some $k \in \{0, 1, \ldots, d-1\}$. It has been shown (for a brief account, see the paper [276]) by Du, Cao and Hsu, that $D_I(d,n)$ are of (almost) optimal diameter and vertex-strong connectivity.

Du, Hsu and Hwang [278] suggested a concept of digraphs extending both the generalized de Bruijn digraphs and the Imase-Ito digraphs. Let d, n be two natural numbers such that $d < n$. Given $q \in [n-1]$ and $r \in \{0, 1, \ldots, n-1\}$, a **consecutive-$d$ digraph** $D(d,n,q,r)$ is the directed pseudograph with vertex set $\{0, 1, \ldots, n-1\}$ such that $i \rightarrow j$ if and only if $j \equiv qi + r + k \pmod{n}$ for some $k \in \{0, 1, \ldots, d-1\}$. Several results on diameter, vertex- and arc-strong connectivity and other properties of consecutive-d digraphs are given in [276]. In Section 6.9, we provide results on hamiltonicity of consecutive-d digraphs.

2.6 Series-Parallel Digraphs

In this section we study vertex series-parallel digraphs and arc series-parallel directed multigraphs. Vertex series-parallel digraphs were introduced by Lawler [637] and Monma and Sidney [701] as a model for scheduling problems. While vertex series-parallel digraphs continue to play an important role for the design of efficient algorithms in scheduling and sequencing problems, they have been extensively studied in their own right as well as in relations to other optimization problems (cf. the papers [55] by Baffi and Petreschi, [153] by Bertolazzi, Cohen, Di Battista, Tamassia and Tollis, [776] by Rendl and [832] by Steiner). Arc series-parallel directed multigraphs were introduced even earlier (than vertex series-parallel digraphs) by Duffin [281] as a mathematical model of electrical networks.

For an acyclic digraph D, let F_D (I_D) be the set of vertices of D of out-degree (in-degree) zero. To define vertex series-parallel digraphs, we first introduce **minimal vertex series-parallel (MVSP) digraphs** recursively.

The digraph of order one with no arc is an MVSP digraph. If $D = (V, A)$, $H = (U, B)$ is a pair of MVSP digraphs ($U \cap V = \emptyset$), so are the acyclic digraphs constructed by each of the following operations (see Figure 2.6):

(a) **Parallel composition:** $P = (V \cup U, A \cup B)$;
(b) **Series composition:** $S = (V \cup U, A \cup B \cup (F_D \times I_H))$.

It is interesting to note that we can embed every MVSP digraph D into the Cartesian plane such that if vertices u, v have coordinates (x_u, y_u) and (x_v, y_v), respectively, then there is a (u, v)-path in D if and only if $x_u \le x_v$ and $y_u \le y_v$. The proof of this non-difficult fact is given in the paper [883] by Valdes, Tarjan and Lawler; see Exercise 2.17. See also Figure 2.8.

An acyclic digraph D is a **vertex series-parallel (VSP)** digraph if the transitive reduction of D is an MVSP digraph (see Section 2.3 for the definition of the transitive reduction). See Figure 2.7.

The following class of acyclic directed multigraphs, **arc series-parallel (ASP) directed multigraphs**, is related to VSP digraphs. The digraph $\vec{P}_2$ is an ASP directed multigraph. If D_1, D_2 is a pair of ASP directed multigraphs with $V(D_1) \cap V(D_2) = \emptyset$, then so are acyclic directed multigraphs constructed by each of the following operations (see Figure 2.9):

(a) **Two-terminal parallel composition:** Choose a vertex u_i of out-degree zero in D_i and a vertex v_i of in-degree zero in D_i for $i = 1, 2$. Identify u_1 with u_2 and v_1 with v_2;
(b) **Two-terminal series composition:** Choose $u \in F_{D_1}$ and $v \in I_{D_2}$ and identify u with v.

Observe that every ASP directed multigraph has a unique vertex of out-degree zero and a unique vertex of in-degree zero. We refer the reader to the book [127] by Battista, Eades, Tamassia and Tollis for several algorithms for drawing graphs nicely, in particular drawing of ASP digraphs.

The next result shows a relation between the classes of digraphs introduced above.

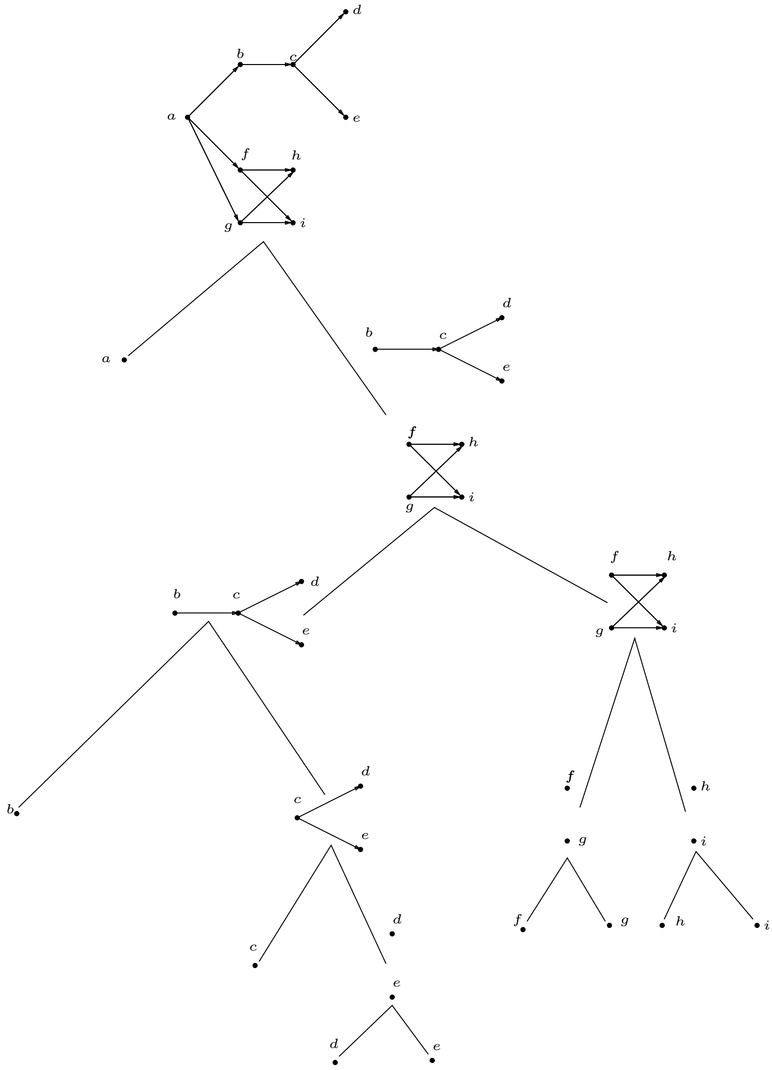


Figure 2.6 (De)construction of an MVSP digraph R_0 by series and parallel (de)compositions.

Theorem 2.6.1 *An acyclic directed multigraph D with a unique vertex of out-degree zero and a unique vertex of in-degree zero is ASP if and only if $L(D)$ is an MVSP digraph.*

Proof: This can be proved easily by induction on $|A(D)|$ using the following two facts:

(i) $L(\vec{P}_2) = \vec{P}_1$, which is an MVSP digraph;

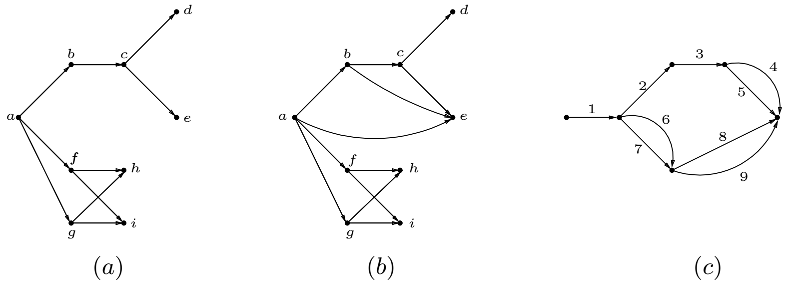


Figure 2.7 Series-parallel directed multigraphs: (a) an MVSP digraph R_0, (b) a VSP digraph R_1, (c) an ASP directed multigraph H_0.

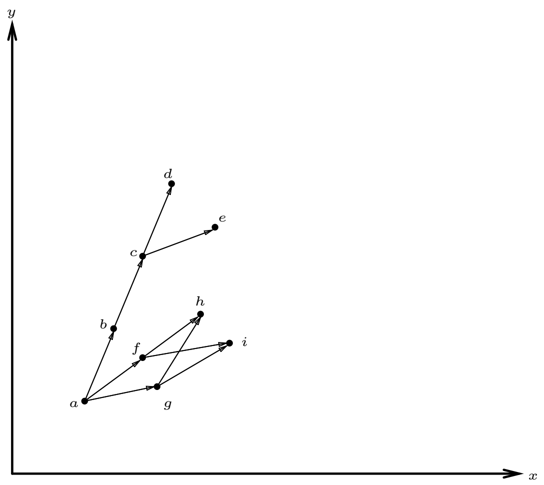


Figure 2.8 The MVSP digraph R_0 of Figure 2.6 embedded into the Cartesian plane such that for every (u,v)-path in R_0 we have $x_u \le x_v$ and $y_u \le y_v$ (and vice versa).

(ii) The line digraph of the two-terminal series (parallel) composition of D_1 and D_2 is the series (parallel) composition of $L(D_1)$ and $L(D_2)$. □

It is easy to check that $L(H_0) = R_0$ for directed multigraphs H_0 and R_0 depicted in Figure 2.7. The following operations in a directed multigraph D are called **reductions:**

(a) **Series reduction:** Replace a path uvw, where $d^+_D(v) = d^-_D(v) = 1$ by the arc uw;
(b) **Parallel reduction:** Replace a pair of parallel arcs from u to v by just one arc from u to v.

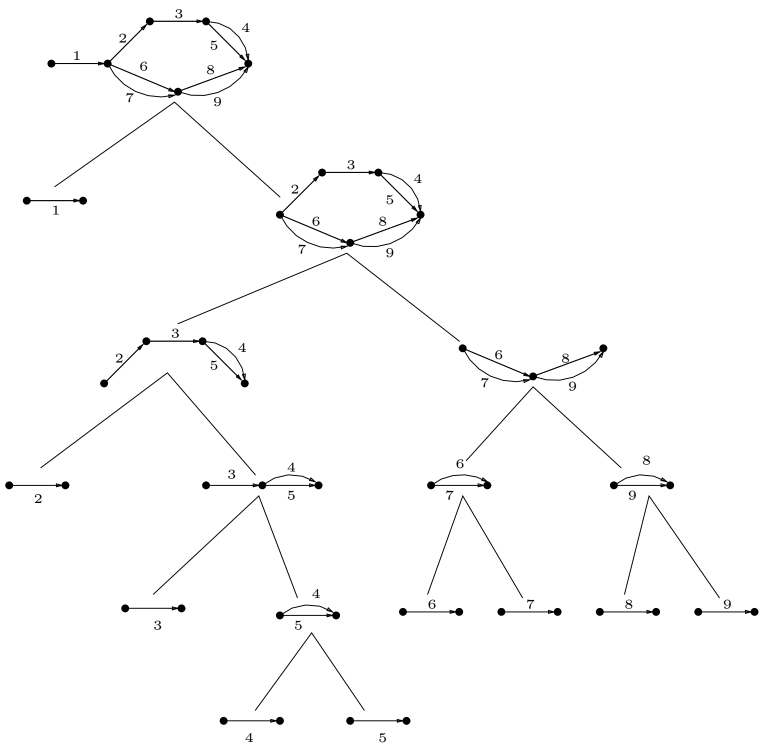


Figure 2.9 (De)construction of an ASP directed multigraph H_0 by two-terminal series and parallel (de)compositions.

The following proposition due to Duffin (see also the paper [883] by Valdes, Tarjan and Lawler) gives a characterization of ASP directed multigraphs. Its proof is left as Exercise 2.18.

Proposition 2.6.2 *[281] A directed multigraph is ASP if and only if it can be reduced to $\vec{P}_2$ by a sequence of series and parallel reductions.* □

The reader is advised to apply a sequence of series and parallel reductions to the directed multigraph H_0 of Figure 2.7 to obtain a digraph isomorphic to $\vec{P}_2$. From the algorithmic point of view, it is important that *every* sequence of series and parallel reductions transforms a directed multigraph to the same digraph. Indeed, this implies an obvious polynomial algorithm to verify if a given directed multigraph is ASP. The proof of the following result, due to Harary, Krarup and Schwenk, is left as Exercise 2.19.

Proposition 2.6.3 *[500] For every acyclic directed multigraph D, the result of application of series and parallel reductions until one can apply such reductions is a unique digraph H.* □

In [883], Valdes, Tarjan and Lawler showed how to construct a linear-time algorithm to recognize ASP directed multigraphs, which is based on Propositions 2.6.2 and 2.6.3. They also presented a more complicated linear-time algorithm to recognize VSP digraphs. Since we are limited in space, we will not discuss the details of the linear-time algorithms. Instead, we will consider the following simplified polynomial algorithm to recognize VSP digraphs.

VSP recognition algorithm
Input: An acyclic digraph D.
Output: YES if D is VSP and NO, otherwise.

1. Compute the transitive reduction R of D.
2. Try to compute an acyclic directed multigraph H with $|I_H| = |F_H| = 1$ such that $L(H) = R$. If there is no such H, then output NO.
3. Verify whether H is an ASP directed multigraph. If it is so, then YES, otherwise, NO.

We prove first the correctness of this algorithm. If the output is YES, then, by Theorem 2.6.1, R is MVSP and thus D is VSP. If H in Step 2 is not found, then, by Theorem 2.6.1, R is not MVSP implying that D is not VSP. If H is not ASP, then R is not MVSP by the same theorem.

Now we prove that the algorithm is polynomial. Step 1 can be performed in polynomial time by Proposition 2.3.5. Step 2 can be implemented using Procedure Build-H described at the end of Section 2.4. This procedure implies that if there is an H such that $L(H) = R$, then there is such an H with additional property that $|I_H| = |F_H| = 1$. The procedure is polynomial. Finally, Step 3 is polynomial by the remark after Proposition 2.6.2.

2.7 Quasi-Transitive Digraphs

A digraph D is **quasi-transitive** if, for every triple x, y, z of distinct vertices of D such that xy and yz are arcs of D, there is at least one arc between x and z. Clearly, a semicomplete digraph is quasi-transitive. Note that if there is only one arc between x and z, it can have any direction; hence quasi-transitive digraphs are generally not transitive.

The aim of this section is to derive a recursive characterization of quasi-transitive digraphs which allows one to show that a number of problems for quasi-transitive digraphs including the longest path and cycle problems are polynomial time solvable (see Sections 6.7 and 6.8). The characterization implies that every quasi-transitive digraph is totally Ψ-decomposable, where Ψ is the union of all transitive digraphs and all extended semicomplete digraphs. Our presentation is based on the paper [103] by Bang-Jensen and Huang.

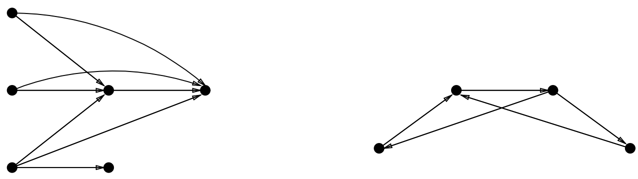


Figure 2.10 A transitive digraph T and a quasi-transitive digraph Q.

An (x_1, x_n)-path $P = x_1x_2 \dots x_n$ is **minimal** if, for every (x_1, x_n)-path Q, either $V(P) = V(Q)$ or Q has a vertex not in $V(P)$.

Proposition 2.7.1 *Let D be a quasi-transitive digraph. Suppose that $P = x_1x_2 \dots x_k$ is a minimal (x_1, x_k)-path. Then the subdigraph induced by $V(P)$ is a semicomplete digraph and $x_j \rightarrow x_i$ for every $2 \leq i+1 < j \leq k$, unless $k = 4$, in which case the arc between x_1 and x_k may be absent.*

Proof: The cases $k = 2, 3, 4, 5$ are easily verified. As an example, let us consider the case $k = 5$. If x_i and x_j are adjacent and $2 \leq i+1 < j \leq 5$, then $x_j \rightarrow x_i$ since P is minimal. Since D is quasi-transitive, x_i and x_{i+2} are adjacent for $i = 1, 2, 3$. This and the minimality of P imply that $x_3 \rightarrow x_1, x_4 \rightarrow x_2$ and $x_5 \rightarrow x_3$. From these arcs and the minimality of P we conclude that $x_5 \rightarrow x_1$. Now the arcs x_4x_5 and x_5x_1 imply that $x_4 \rightarrow x_1$. Similarly, $x_5 \rightarrow x_1 \rightarrow x_2$ implies $x_5 \rightarrow x_2$.

The proof for the case $k \geq 6$ is by induction on k with the case $k = 5$ as the basis. By induction, each of $D\langle\{x_1, x_2, \dots, x_{k-1}\}\rangle$ and $D\langle\{x_2, x_3, \dots, x_k\}\rangle$ is a semicomplete digraph and $x_j \rightarrow x_i$ for any $1 < j - i \leq k - 2$. Hence x_3 dominates x_1 and x_k dominates x_3 and the minimality of P implies that x_k dominates x_1. □

Corollary 2.7.2 *If a quasi-transitive digraph D has an (x, y)-path but x does not dominate y, then either $y \rightarrow x$, or there exist vertices $u, v \in V(D) - \{x, y\}$ such that $x \rightarrow u \rightarrow v \rightarrow y$ and $y \rightarrow u \rightarrow v \rightarrow x$.*

Proof: This is easy to deduce by considering a minimal (x, y)-path and applying Proposition 2.7.1. □

Lemma 2.7.3 *Suppose that A and B are distinct strong components of a quasi-transitive digraph D with at least one arc from A to B. Then $A \mapsto B$.*

Proof: Suppose A and B are distinct strong components such that there exists an arc from A to B. Then for every choice of $x \in A$ and $y \in B$ there exists a path from x to y in D. Since A and B are distinct strong components, none of the alternatives in Corollary 2.7.2 can hold and hence $x \rightarrow y$. □

Lemma 2.7.4 *[103] Let D be a strong quasi-transitive digraph on at least two vertices. Then the following holds:*

(a) $\overline{UG(D)}$ is disconnected;
(b) If S and S' are two subdigraphs of D such that $\overline{UG(S)}$ and $\overline{UG(S')}$ are distinct connected components of $\overline{UG(D)}$, then either $S \mapsto S'$ or $S' \mapsto S$, or both $S \rightarrow S'$ and $S' \rightarrow S$ in which case $|V(S)| = |V(S')| = 1$.

Proof: The statement (b) can be easily verified from the definition of a quasi-transitive digraph and the fact that S and S' are completely adjacent in D (Exercise 2.20). We prove (a) by induction on $|V(D)|$. Statement (a) is trivially true when $|V(D)| = 2$ or 3. Assume that it holds when $|V(D)| < n$ where $n > 3$.

Suppose that there is a vertex z such that $D - z$ is not strong. Then there is an arc from (to) every terminal (initial) component of $D - z$ to (from) z. Since D is quasi-transitive, the last fact and Lemma 2.7.3 imply that $X \rightarrow Y$ for every initial (terminal) strong component X (Y) of $D - z$. Similar arguments show that each strong component of $D - z$ either dominates some terminal component or is dominated by some initial component of $D - z$ (intermediate strong components satisfy both). These facts imply that z is adjacent to every vertex in $D - z$. Therefore, $\overline{UG(D)}$ contains a component consisting of the vertex z, implying that $\overline{UG(D)}$ is disconnected and (a) follows.

Assume that there is a vertex v such that $D - v$ is strong. Since D is strong, D contains an arc vw from v to $D - v$. By induction, $\overline{UG(D - v)}$ is not connected. Let connected components S and S' of $\overline{UG(D - v)}$ be chosen such that $w \in S$, $S \mapsto S'$ in D (here we use (b) and the fact that $D - v$ is strong). Then v is completely adjacent to S' in D (as $v \rightarrow w$). Hence $\overline{UG(S')}$ is a connected component of $\overline{UG(D)}$ and the proof is complete. □

The following theorem completely characterizes quasi-transitive digraphs in recursive sense (see also Figure 2.11).

Theorem 2.7.5 (Bang-Jensen and Huang) *[103] Let D be a digraph which is quasi-transitive.*

(a) If D is not strong, then there exist a transitive oriented graph T with vertices $\{u_1, u_2, \ldots, u_t\}$ and strong quasi-transitive digraphs $H_1, H_2, \ldots, H_t$ such that $D = T[H_1, H_2, \ldots, H_t]$, where H_i is substituted for u_i, $i = 1, 2, \ldots, t$.
(b) If D is strong, then there exists a strong semicomplete digraph S with vertices $\{v_1, v_2, \ldots, v_s\}$ and quasi-transitive digraphs $Q_1, Q_2, \ldots, Q_s$ such that Q_i is either a vertex or is non-strong and $D = S[Q_1, Q_2, \ldots, Q_s]$, where Q_i is substituted for v_i, $i = 1, 2, \ldots, s$.

Proof: Suppose that D is not strong and let $H_1, H_2, \ldots, H_t$ be the strong components of D. According to Lemma 2.7.3, if there is an arc between

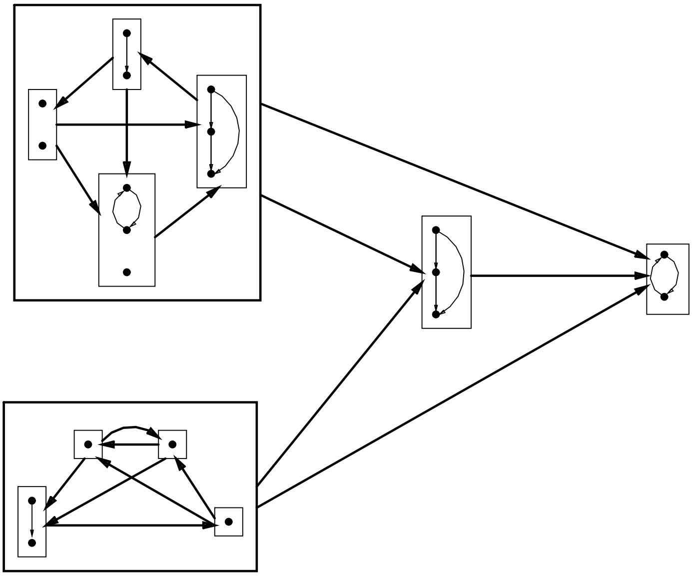


Figure 2.11 A decomposition of a non-strong quasi-transitive digraph. Big arcs between different boxed sets indicate that there is a complete domination in the direction shown.

H_i and H_j, then either $H_i \mapsto H_j$ or $H_j \mapsto H_i$. Now if $H_i \mapsto H_j \mapsto H_k$, then, by quasi-transitivity, $H_i \mapsto H_k$. So by contracting each H_i to a vertex h_i, we get a transitive oriented graph T with vertices $h_1, h_2, \ldots, h_t$. This shows that $D = T[H_1, H_2, \ldots, H_t]$.

Suppose now that D is strong. Let $Q_1, Q_2, \ldots, Q_s$ be the subdigraphs of D such that each $\overline{UG(Q_i)}$ is a connected component of $\overline{UG(D)}$. According to Lemma 2.7.4(a), each Q_i is either non-strong or just a single vertex. By Lemma 2.7.4(b) we obtain a strong semicomplete digraph S if each Q_i is contracted to a vertex. This shows that $D = S[Q_1, Q_2, \ldots, Q_s]$. □

2.8 Path-Mergeable Digraphs

A digraph D is **path-mergeable**, if for any choice of vertices $x, y \in V(D)$ and any pair of internally disjoint (x, y)-paths P, Q, there exists an (x, y)-path R in D, such that $V(R) = V(P) \cup V(Q)$. We will see, in several places of this book, that the notion of a path-mergeable digraph is very useful for design of algorithms and proofs of theorems. This makes it worthwhile studying path-mergeable digraphs. The results presented in this section are adapted from [72], where the study of path-mergeable digraphs was initiated by Bang-Jensen.

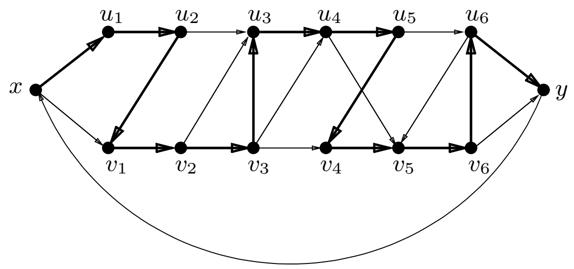


Figure 2.12 A digraph which is path-mergeable. The fat arcs indicate the path $xu_1u_2v_1v_2v_3u_3u_4u_5v_4v_5v_6u_6y$ from x to y which is obtained by merging the two (x,y)-paths $xu_1u_2u_3u_4u_5u_6y$ and $xv_1v_2v_3v_4v_5v_6y$.

We prove a characterization of path-mergeable digraphs, which implies that path-mergeable digraphs can be recognized efficiently.

Theorem 2.8.1 *A digraph D is path-mergeable if and only if for every pair of distinct vertices $x,y \in V(D)$ and every pair $P = xx_1 \ldots x_ry$, $P' = xy_1 \ldots y_sy$, $r,s \geq 1$ of internally disjoint (x,y)-paths in D, either there exists an $i \in \{1,\ldots,r\}$, such that $x_i {\rightarrow} y_1$, or there exists a $j \in [s]$, such that $y_j {\rightarrow} x_1$.*

Proof: We prove 'only if' by induction on $r+s$. It is obvious for $r = s = 1$, so suppose that $r + s \geq 3$. If there is no arc between $\{x_1,\ldots,x_r\}$ and $\{y_1,\ldots,y_s\}$, then clearly P, P' cannot be merged into one path. Hence we may assume without loss of generality that there is an arc x_iy_j for some i,j, $1 \leq i \leq r, 1 \leq j \leq s$. If $j = 1$, then the claim follows. Otherwise apply induction to the paths $P[x,x_i]y_j$, $xP'[y_1,y_j]$.

The proof of 'if' is left to the reader. It is similar to the proof of Proposition 2.8.3 below. □

The proof of the following result is left as Exercise 2.24.

Corollary 2.8.2 *Path-mergeable digraphs can be recognized in polynomial time.* □

The next result shows that if a digraph is path-mergeable, then the merging of paths can always be done in a particularly nice way.

Proposition 2.8.3 *Let D be a digraph which is path-mergeable and let $P = xx_1 \ldots x_ry$, $P' = xy_1 \ldots y_sy$, $r,s \geq 0$ be internally disjoint (x,y)-paths in D. The paths P and P' can be merged into one (x,y)-path P^* such that vertices from P (respectively, P') remain in the same order as on that path. Furthermore the merging can be done in at most $2(r+s)$ steps.*

Proof: We prove the result by induction on $r+s$. It is obvious if $r = 0$ or $s = 0$, so suppose that $r,s \geq 1$. By Theorem 2.8.1 there exists an i such that either $x_i \to y_1$ or $y_i \to x_1$. By scanning both paths forward one arc at a time, we can find i in at most $2i$ steps; suppose without loss of generality $x_i \to y_1$. By applying the induction hypothesis to the paths $P[x_i, x_r]y$ and $x_i P'[y_1, y_s]y$, we see that we can merge them into a single path Q in the required order-preserving way in at most $2(r+s-i)$ steps. The required path P^* is obtained by concatenating the paths $xP[x_1, x_i]$ and Q, and we have found it in at most $2(r+s)$ steps, as required. □

2.9 Locally In/Out-Semicomplete Digraphs

A digraph D is **locally in-semicomplete** (**locally out-semicomplete**) if, for every vertex x of D, the in-neighbours (out-neighbours) of x induce a semicomplete digraph. Clearly, the converse of a locally in-semicomplete digraph is a locally out-semicomplete digraph and vice versa. A digraph D is **locally semicomplete** if it is both locally in- and locally out-semicomplete. See Figure 2.13. Clearly every semicomplete digraph is locally semicomplete. A locally in-semicomplete digraph with no 2-cycle is a **locally in-tournament digraph**. Similarly, one can define **locally out-tournament digraphs** and **locally tournament digraphs**. For convenience, we will sometimes refer to locally tournament digraphs as **local tournaments** and to locally in-tournament (out-tournament) digraphs as **local in-tournaments** (**local out-tournaments**).

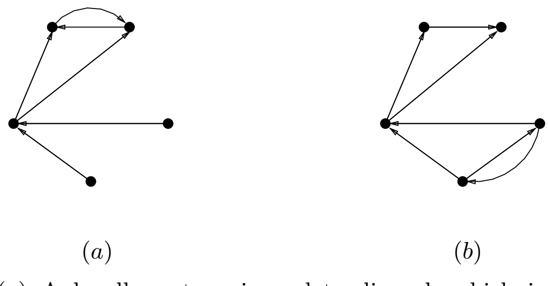


Figure 2.13 (a) A locally out-semicomplete digraph which is not locally in-semicomplete; (b) a locally semicomplete digraph.

Proposition 2.9.1 by Bang-Jensen shows that locally in-semicomplete and locally out-semicomplete digraphs form subclasses of the class of path-mergeable digraphs. In particular, this means that every tournament is path-mergeable. In many theorems and algorithms on tournaments this property is of essential use. In some other cases, the very use of this property allows one to simplify proofs of results on tournaments and their generalizations or speed up algorithms on those digraphs.

Proposition 2.9.1 *[72] Every locally in-semicomplete (out-semicomplete) digraph is path-mergeable.*

Proof: Let D be a locally out-semicomplete digraph and let $P = y_1y_2 \ldots y_k$, $Q = z_1z_2 \ldots z_t$ be a pair of internally disjoint (x, y)-paths (i.e., $y_1 = z_1 = x$ and $y_k = z_t = y$). We show that there exists an (x, y)-path R in D, such that $V(R) = V(P) \cup V(Q)$. Our claim is trivially true when $|A(P)| + |A(Q)| = 3$. Assume now that $|A(P)| + |A(Q)| \geq 4$. Since D is out-semicomplete, either $y_2 \rightarrow z_2$ or $z_2 \rightarrow y_2$ (or both) and the claim follows from Theorem 2.8.1.

The proposition holds for locally in-semicomplete digraphs as they are the converses of locally out-semicomplete digraphs. □

The path-mergeability can be generalized in a natural way as follows. A digraph D is **in-path-mergeable** if, for every vertex $y \in V(D)$ and every pair P, Q of internally disjoint paths with common terminal vertex y, there is a path R such that $V(R) = V(P) \cup V(Q)$, the path R terminates at y and starts at a vertex which is the initial vertex of either P or Q (or, possibly, both). Observe that, in this definition, the initial vertices of paths P and Q may coincide. Therefore, every in-path-mergeable digraph is path-mergeable. However, it is easy to see that not every path-mergeable digraph is in-path-mergeable (see Exercise 2.21). A digraph D is **out-path-mergeable** if the converse of D is in-path-mergeable. Clearly, every in-path-mergeable (out-path-mergeable) digraph is locally in-semicomplete (locally out-semicomplete). The converse is also true (hence this is another way of characterizing locally in-semicomplete digraphs). The proof of Proposition 2.9.2 is left as Exercise 2.25.

Proposition 2.9.2 *Every locally in-semicomplete (out-semicomplete, respectively) digraph is in-path-mergeable (out-path-mergeable, respectively).* □

Some simple, yet very useful, properties of locally in-semicomplete digraphs are described in the following results (in [105], by Bang-Jensen, Huang and Prisner, these results were proved for locally tournament digraphs only, so the statements below are their slight generalizations first stated by Bang-Jensen and Gutin [89]). Observe that a locally out-semicomplete digraph, being the converse of a locally in-semicomplete digraph, has similar properties (see Exercise 2.28). The next lemma follows from Proposition 1.7.1 (see [91]).

Lemma 2.9.3 *Every connected locally in-semicomplete digraph D has an out-branching.* □

Theorem 2.9.4 is illustrated in Figure 2.14.

Theorem 2.9.4 *Let D be a locally in-semicomplete digraph.*

(i) Let A and B be distinct strong components of D. If a vertex $a \in A$ dominates some vertex in B, then $a \mapsto B$.
(ii) If D is connected, then $SC(D)$ has an out-branching.

Proof: Let A and B be strong components of D for which there is an arc (a, b) from A to B. Since B is strong, there is a (b', b)-path in B for every $b' \in V(B)$. By the definition of locally in-semicomplete digraphs and the fact that there is no arc from B to A, we can conclude that $a \rightarrow b'$. This proves (i).

Part (ii) follows from the fact that $SC(D)$ is itself a locally in-tournament digraph and Lemma 2.9.3. □

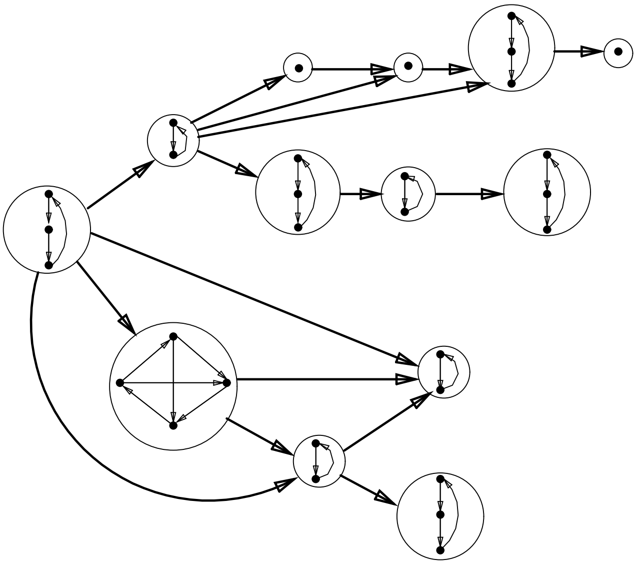


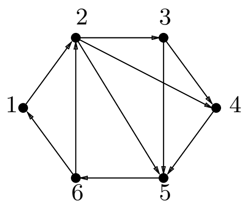
Figure 2.14 The strong decomposition of a non-strong locally in-semicomplete digraph. The big circles indicate strong components and a fat arc from a component A to a component B between two components indicates that there is at least one vertex $a \in A$ such that $a \mapsto B$.

2.10 Locally Semicomplete Digraphs

Locally semicomplete digraphs were introduced in 1990 by Bang-Jensen [66]. As shown in several places in our book, this class of digraphs has many nice properties in common with its proper subclass, semicomplete digraphs. The main aim of this section is to obtain a classification of locally semicomplete digraphs first proved by Bang-Jensen, Guo, Gutin and Volkmann [80]. In the process of deriving this classification, we will show several important properties of locally semicomplete digraphs. We start our consideration from round digraphs, a nice special class of locally semicomplete digraphs.

2.10.1 Round Digraphs

A digraph on n vertices is **round** if we can label its vertices $v_1, v_2, \ldots, v_n$ so that for each i, we have $N^+(v_i) = \{v_{i+1}, \ldots, v_{i+d^+(v_i)}\}$ and $N^-(v_i) = \{v_{i-d^-(v_i)}, \ldots, v_{i-1}\}$ (all subscripts are taken modulo n). We will refer to the ordering $v_1, v_2, \ldots, v_n$ as a **round labelling** of D. See Figure 2.15 for an example of a round digraph. Observe that every strong round digraph D is hamiltonian, since $v_1 v_2 \ldots v_n v_1$ form a hamiltonian cycle, whenever $v_1, v_2, \ldots, v_n$ is a round labelling. Round digraphs form a subclass of locally semicomplete digraphs. We will see below that round digraphs play an important role in the study of locally semicomplete digraphs.



R

Figure 2.15 A round digraph with a round labelling.

Proposition 2.10.1 *[541] Every round digraph is locally semicomplete.*

Proof: Let D be a round digraph and let $v_1, v_2, \ldots, v_n$ be a round labelling of D. Consider an arbitrary vertex, say v_i. Let x, y be a pair of out-neighbours of v_i. We show that x and y are adjacent. Assume without loss of generality that v_i, x, y appear in that circular order in the round labelling. Since $v_i \rightarrow y$ and the in-neighbours of y appear consecutively preceding y, we must have $x \rightarrow y$. Thus the out-neighbours of v_i are pairwise adjacent. Similarly, we can show that the in-neighbours of v_i are also pairwise adjacent. Therefore, D is locally semicomplete. $\square$

The main result of this subsection is Theorem 2.10.4 of Huang [541] that gives a characterization of round locally semicomplete digraphs. This characterization generalizes the corresponding characterizations of round local tournaments and tournaments, due to Bang-Jensen [66] and Alspach and Tabib [38], respectively.

An arc xy of a digraph D is **ordinary** if yx is not in D. A cycle or path Q of a digraph D is **ordinary** if all arcs of Q are ordinary.

The following two lemmas due to Huang [541] imply the necessity part of Theorem 2.10.4. A sufficiency proof can be found in [91, 541].

Lemma 2.10.2 *Let D be a round digraph; then the following is true:*

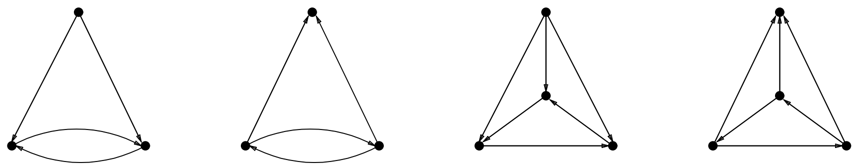


Figure 2.16 Some forbidden digraphs in Huang's characterization.

(a) Every induced subdigraph of D is round.
(b) None of the digraphs in Figure 2.16 is an induced subdigraph of D.
(c) For each $x \in V(D)$, the subdigraphs induced by $N^+(x) - N^-(x)$ and $N^-(x) - N^+(x)$ are transitive tournaments.

Proof: Exercise 2.31. □

Lemma 2.10.3 *Let D be a round digraph. Then, for each vertex x of D, the subdigraph induced by $N^+(x) \cap N^-(x)$ contains no ordinary cycle.*

Proof: Suppose the subdigraph induced by some $N^+(x) \cap N^-(x)$ contains an ordinary cycle C. Let $v_1, v_2, \ldots, v_n$ be a round labelling of D. Without loss of generality, assume that $x = v_1$. Then C must contain an arc v_iv_j such that $v_jv_i \notin A(D)$ and $i > j$. We have $v_1 \in N^-(v_i)$ but $v_j \notin N^-(v_i)$, contradicting the assumption that $v_1, v_2, \ldots, v_n$ is a round labelling of D. □

Theorem 2.10.4 (Huang) *[541] A connected locally semicomplete digraph D is round if and only if the following holds for each vertex x of D:*

(a) $N^+(x) - N^-(x)$ and $N^-(x) - N^+(x)$ induce transitive tournaments and
(b) $N^+(x) \cap N^-(x)$ induces a (semicomplete) subdigraph containing no ordinary cycle. □

The proof of sufficiency of the conditions of this theorem in [91, 541] can be transformed into a polynomial time algorithm to decide whether a digraph D is round and to find a round labelling of D (if D is round).

Corollary 2.10.5 (Bang-Jensen) *[66] A connected local tournament D is round if and only if, for each vertex x of D, $N^+(x)$ and $N^-(x)$ induce transitive tournaments.* □

2.10.2 Non-Strong Locally Semicomplete Digraphs

The most basic properties of strong components of a connected non-strong locally semicomplete digraph are given in the following result, due to Bang-Jensen.

Theorem 2.10.6 *[66] Let D be a connected locally semicomplete digraph that is not strong. Then the following holds for D.*

(a) If A and B are distinct strong components of D with at least one arc between them, then either $A\mapsto B$ or $B\mapsto A$.
(b) If A and B are strong components of D, such that $A\mapsto B$, then A and B are semicomplete digraphs.
(c) The strong components of D can be ordered in a unique way $D_1, D_2, \dots, D_p$ such that there are no arcs from D_j to D_i for $j > i$, and D_i dominates D_{i+1} for $i \in [p-1]$.

Proof: Recall that a locally semicomplete digraph is a locally in-semicomplete digraph as well as a locally out-semicomplete digraph. Part (a) of this theorem follows immediately from Part (i) of Theorem 2.9.4 and its analogue for locally out-semicomplete digraphs. Part (b) can be easily obtained from the definition of a locally semicomplete digraph. Finally, Part (c) follows from the fact proved in Theorem 2.9.4 (and its analogue for locally out-semicomplete digraphs) that $SC(D)$ has an out-branching and an in-branching. Indeed, a digraph which is both out-branching and in-branching is merely a hamiltonian path. □

A locally semicomplete digraph D is **round decomposable** if there exists a round local tournament R on $r \geq 2$ vertices such that $D = R[S_1, \dots, S_r]$, where each S_i is a strong semicomplete digraph. We call $R[S_1, \dots, S_r]$ a **round decomposition** of D. The following consequence of Theorem 2.10.6, whose proof is left as Exercise 2.32, shows that connected, but not strongly connected locally semicomplete digraphs are round decomposable.

Corollary 2.10.7 *[66] Every connected, but not strongly connected locally semicomplete digraph D has a unique round decomposition $R[D_1, D_2, \dots, D_p]$, where $D_1, D_2, \dots, D_p$ is the acyclic ordering of strong components of D and R is the round local tournament containing no cycle which one obtains by taking one vertex from each D_i.* □

Now we describe another kind of decomposition theorem for locally semicomplete digraphs due to Guo and Volkmann. The proof of this theorem is left as Exercise 2.33. The statement of the theorem is illustrated in Figure 2.18.

Theorem 2.10.8 *[440, 442] Let D be a connected locally semicomplete digraph that is not strong and let $D_1, \dots, D_p$ be the acyclic ordering of strong components of D. Then D can be decomposed into $r \geq 2$ induced subdigraphs $D'_1, D'_2, \dots, D'_r$ as follows:*

- $D'_1 = D_p, \quad \lambda_1 = p,$
- $\lambda_{i+1} = \min\{\, j \mid N^+(D_j) \cap V(D'_i) \neq \emptyset\}$, *for each $i \in [r-1]$,*

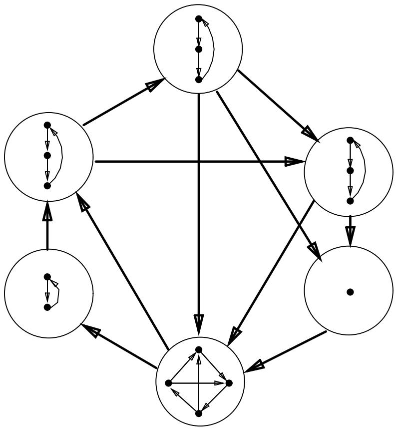


Figure 2.17 A round decomposable locally semicomplete digraph D. The big circles indicate the sets that correspond to the sets $W_1, W_2, \ldots, W_6$ in the decomposition $D = R[W_1, W_2, \ldots, W_6]$, where R is the round locally semicomplete digraph one obtains by replacing each circled set by one vertex. Fat arcs indicate that there is a complete domination in the direction shown.

- $D'_{i+1} = D\langle V(D_{\lambda_{i+1}}) \cup V(D_{\lambda_{i+1}+1}) \cup \cdots \cup V(D_{\lambda_i - 1})\rangle$, *for each* $i \in [r-1]$.

The subdigraphs $D'_1, D'_2, \ldots, D'_r$ *satisfy the properties below:*

(a) D'_i *consists of some strong components of* D *and is semicomplete for each* $i \in [r]$

(b) D'_{i+1} *dominates the initial component of* D'_i *and there exists no arc from* D'_i *to* D'_{i+1} *for any* $i \in [r-1]$

(c) if $r \geq 3$*, then there is no arc between* D'_i *and* D'_j *for* i, j *satisfying* $|j-i| \geq 2$. □

For a connected, but not strongly connected locally semicomplete digraph D, the unique sequence $D'_1, D'_2, \ldots, D'_r$ defined in Theorem 2.10.8 is called the **semicomplete decomposition** of D.

2.10.3 Strong Round Decomposable Locally Semicomplete Digraphs

In the previous subsection we saw that every connected non-strong locally semicomplete digraph is round decomposable. This property does not hold for strong locally semicomplete digraphs (see Lemma 2.10.14). The following assertions, due to Bang-Jensen, Guo, Gutin and Volkmann, provide some important properties concerning round decompositions of strong locally semicomplete digraphs.

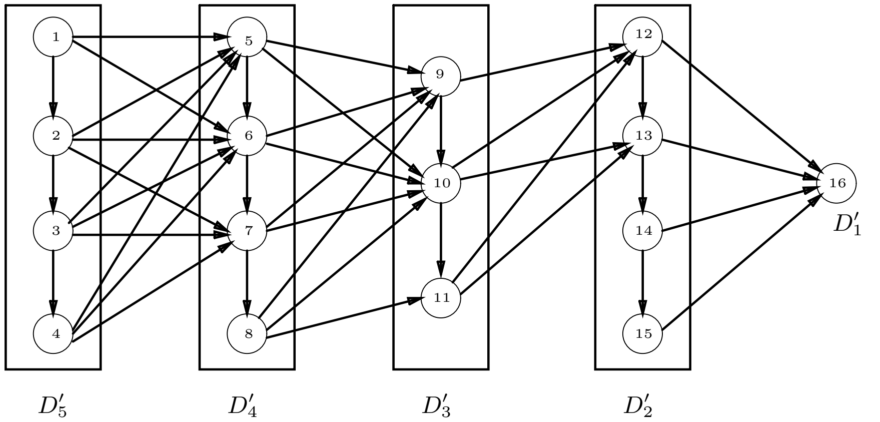


Figure 2.18 The semicomplete decomposition of a non-strong locally semicomplete digraph with 16 strong components (numbered 1-16 corresponding to the acyclic ordering). Each circle indicates a strong component and each box indicates a semicomplete subdigraph formed by consecutive components all of which dominate the first component in the previous layer. For clarity arcs inside components as well as some arcs between components inside a semicomplete subdigraph D'_i (all going from top to bottom) are omitted.

Proposition 2.10.9 *[80] Let $R[H_1, H_2, \ldots, H_\alpha]$ be a round decomposition of a strong locally semicomplete digraph D. Then, for every minimal separating set S, there are two integers i and $k \geq 0$ such that $S = V(H_i) \cup \ldots \cup V(H_{i+k})$.*

Proof: We will first prove that

$$\text{if } V(H_i) \cap S \neq \emptyset, \text{ then } V(H_i) \subseteq S. \tag{2.3}$$

Assume that there exists H_i such that $V(H_i) \cap S \neq \emptyset \neq V(H_i) - S$. Using this assumption we shall prove that $D - S$ is strong, contradicting the definition of S.

Let $s' \in V(H_i) \cap S$. To show that $D - S$ is strong, we consider a pair of different vertices x and y of $D - S$ and prove that $D - S$ has an (x, y)-path. Since S is a minimal separating set, $D' = D - (S - s')$ is strong. Consider a shortest (x, y)-path P in D' among all (x, y)-paths using at most two vertices from each H_j. The existence of such a path follows from the fact that R is strong. Since the vertices of H_i in D' have the same in- and out-neighbourhoods, P contains at most one vertex from H_i, unless $x, y \in V(H_i)$ in which case P contains only these two vertices from H_i. If s' is not on P, we are done. Thus, assume that s' is on P. Then, since P is shortest possible, neither x nor y belongs to H_i. Now we can replace s' with a vertex in $V(H_i) - S$. Therefore, $D - S$ has an (x, y)-path, so (2.3) is proved.

Suppose that S consists of disjoint sets $T_1, \ldots, T_\ell$ such that

$$T_i = V(H_{j_i}) \cup \ldots \cup V(H_{j_i+k_i}) \quad \text{and} \quad (V(H_{j_i-1}) \cup V(H_{j_i+k_i+1})) \cap S = \emptyset$$

for $i \in [\ell]$. If $\ell \geq 2$, then $D - T_i$ is strong and hence it follows from the fact that R is round that H_{j_i-1} dominates $H_{j_i+k_i+1}$ for every $i \in [\ell]$. Therefore, $D - S$ is strong; a contradiction. □

Corollary 2.10.10 *[80] If a locally semicomplete digraph D is round decomposable, then it has a unique round decomposition $D = R[D_1, D_2, \ldots, D_\alpha]$.*

Proof: Suppose that D has two different round decompositions: $D = R[D_1, \ldots, D_\alpha]$ and $D = R'[H_1, \ldots, H_\beta]$.

By Corollary 2.10.7, we may assume that D is strong. By the definition of a round decomposition, this implies that $\alpha, \beta \geq 3$. Let S be a minimal separating set of D. By Proposition 2.10.9, we may assume without loss of generality that $S = V(D_1 \cup \ldots \cup D_i) = V(H_1 \cup \ldots \cup H_j)$ for some i and j. Since $D - S$ is non-strong, by Corollary 2.10.7, $D_{i+1} = H_{j+1}, \ldots, D_\alpha = H_\beta$ (in particular, $\alpha - i = \beta - j$). Now it suffices to prove that

$$D_1 = H_1, \ldots, D_i = H_j \text{ (in particular, } i = j). \tag{2.4}$$

If $D\langle S\rangle$ is non-strong, then (2.4) follows by Corollary 2.10.7. If $D\langle S\rangle$ is strong, then first consider the case $\alpha = 3$. Then $S = V(D_1)$, because $D - S$ is non-strong and $\alpha = 3$. Assuming that $j > 1$, we obtain that the subdigraph of D induced by S has a strong round decomposition. This contradicts the fact that R' is a local tournament, since the in-neighbourhood of the vertex r'_{j+1} in R' contains a cycle (where r'_p corresponds to H_p, $p = 1, \ldots, \beta$). Therefore, (2.4) is true for $\alpha = 3$. If $\alpha > 3$, then we can find a separating set in $D\langle S\rangle$ and conclude by induction that (2.4) holds. □

Proposition 2.10.9 allows us to construct a polynomial algorithm for checking whether a locally semicomplete digraph is round decomposable.

Proposition 2.10.11 *[80] There exists a polynomial algorithm to decide whether a given locally semicomplete digraph D has a round decomposition and to find this decomposition if it exists.*

Proof: We only give a sketch of such an algorithm. Find a minimal separating set S in D starting with $S' = N^+(x)$ for a vertex $x \in V(D)$ and deleting vertices from S' until a minimal separating set is obtained. Construct the strong components of $D\langle S\rangle$ and $D - S$ and label these $D_1, D_2, \ldots, D_\alpha$, where $D_1, \ldots, D_p$, $p \geq 1$, form an acyclic ordering of the strong components of $D\langle S\rangle$ and $D_{p+1}, \ldots, D_\alpha$ form an acyclic ordering of the strong components of $D - S$. For every pair D_i and D_j $(1 \leq i \neq j \leq \alpha)$, we check the following: if there exist some arcs between D_i and D_j, then either $D_i \mapsto D_j$ or $D_j \mapsto D_i$. If we find a pair for which the above condition is false, then D is not round decomposable. Otherwise, we form a digraph $R = D\langle\{x_1, x_2, \ldots, x_\alpha\}\rangle$, where $x_i \in V(D_i)$ for each $i \in [\alpha]$. We check whether R is round using Corollary 2.10.5. If R is not round, then D is not round decomposable. Otherwise, D is round decomposable and $D = R[D_1, \ldots, D_\alpha]$.

It is not difficult to verify that our algorithm is correct and polynomial. □

2.10.4 Classification of Locally Semicomplete Digraphs

We start this subsection with a lemma on minimal separating sets of locally semicomplete digraphs. It will be shown in Lemma 5.8.4 that for a strong locally semicomplete digraph D and a minimal separating set S in D, we have that $D - S$ is connected.

Lemma 2.10.12 *[80] If a strong locally semicomplete digraph D is not semicomplete, then there exists a minimal separating set $S \subset V(D)$ such that $D - S$ is not semicomplete. Furthermore, if $D_1, D_2, \ldots, D_p$ is the acyclic ordering of the strong components of D and $D'_1, D'_2, \ldots, D'_r$ is the semicomplete decomposition of $D - S$, then $r \geq 3$, $D\langle S\rangle$ is semicomplete and we have $D_p \mapsto S \mapsto D_1$.*

Proof: Suppose $D - S$ is semicomplete for every minimal separating set S. Then $D - S$ is semicomplete for all separating sets S. Hence D is semicomplete, because any pair of non-adjacent vertices can be separated by some separating set S. This proves the first claim of the lemma.

Let S be a minimal separating set such that $D - S$ is not semicomplete. Clearly, if $r = 2$ (in Theorem 2.10.8), then $D - S$ is semicomplete. Thus, $r \geq 3$. By the minimality of S every vertex $s \in S$ dominates a vertex in D_1 and is dominated by a vertex in D_p. Thus if some $x \in D_p$ was dominated by $s \in S$, then, by the definition of a locally semicomplete digraph, we would have $D_1 \mapsto D_p$, contradicting the fact that $r \geq 3$. Hence (using that D_p is strongly connected) we get that $D_p \mapsto S$ and similarly $S \mapsto D_1$. From the last observation it follows that S is semicomplete. □

Now we consider strongly connected locally semicomplete digraphs which are not semicomplete and not round decomposable. We first show that the semicomplete decomposition of $D-S$ has exactly three components, whenever S is a minimal separating set such that $D - S$ is not semicomplete.

Lemma 2.10.13 *[80] Let D be a strong locally semicomplete digraph which is not semicomplete. Either D is round decomposable, or D has a minimal separating set S such that the semicomplete decomposition of $D - S$ has exactly three components D'_1, D'_2, D'_3.*

Proof: By Lemma 2.10.12, D has a minimal separating set S such that the semicomplete decomposition of $D - S$ has at least three components.

Assume now that the semicomplete decomposition of $D - S$ has more than three components $D'_1, \ldots, D'_r$ $(r \geq 4)$. Let $D_1, D_2, \ldots, D_p$ be the acyclic ordering of strong components of $D - S$. According to Theorem 2.10.8 (c), there is no arc between D'_i and D'_j if $|i-j| \geq 2$. It follows from the definition of a locally semicomplete digraph that

$$N^+(D'_i) \cap S = \emptyset \text{ for } i \geq 3 \text{ and } N^-(D'_j) \cap S = \emptyset \text{ for } j \leq r-2. \tag{2.5}$$

By Lemma 2.10.12, $D\langle S\rangle$ is semicomplete and $S = N^+(D_p)$. Let $D_{p+1}, \ldots, D_{p+q}$ be the acyclic ordering of the strong components of $D\langle S\rangle$. Using (2.5) and the assumption $r \geq 4$, it is easy to check that if there is an arc between D_i and D_j $(1 \leq i \neq j \leq p+q)$, then $D_i \mapsto D_j$ or $D_j \mapsto D_i$. Let $R = D\langle\{x_1, x_2, \ldots, x_{p+q}\}\rangle$ with $x_i \in V(D_i)$ for each $i \in [p+q]$. Now it suffices to prove that R is a round local tournament.

Since R is a subdigraph of D and no pair D_i, D_j induces a strong digraph, we see that R is a local tournament. By Corollary 2.10.7 each of the subdigraphs $R' = R - \{x_{p+1}, \ldots, x_{p+q}\}$, $R'' = R - V(R) \cap V(D'_{r-1})$ and $R''' = R - V(R) \cap V(D'_2)$ is round. Since $N^+(v) \cap V(R)$ (as well as $N^-(v) \cap V(R)$) is completely contained in one of the sets $V(R'), V(R'')$ and $V(R''')$ for every $v \in V(R)$, we see that R is round.

Thus if $r \geq 4$, then D is round decomposable. □

Our next result is a characterization of locally semicomplete digraphs which are not semicomplete and not round decomposable. This characterization was proved for the first time by Guo in [432]. A weaker form was obtained earlier by Bang-Jensen in [71]. Here we give the proof of this result from [80].

Lemma 2.10.14 *Let D be a strong locally semicomplete digraph which is not semicomplete. Then D is not round decomposable if and only if the following conditions are satisfied:*

(a) There is a minimal separating set S such that $D - S$ is not semicomplete and for each such S, $D\langle S\rangle$ is semicomplete and the semicomplete decomposition of $D - S$ has exactly three components D'_1, D'_2, D'_3;

(b) There are integers α, β, μ, ν with $\lambda_2 \leq \alpha \leq \beta \leq p-1$ and $p+1 \leq \mu \leq \nu \leq p+q$ such that

$$N^-(D_\alpha) \cap V(D_\mu) \neq \emptyset \;\; \textit{and} \;\; N^+(D_\alpha) \cap V(D_\nu) \neq \emptyset,$$

$$\textit{or} \quad N^-(D_\mu) \cap V(D_\alpha) \neq \emptyset \;\; \textit{and} \;\; N^+(D_\mu) \cap V(D_\beta) \neq \emptyset,$$

where $D_1, D_2, \ldots, D_p$ and $D_{p+1}, \ldots, D_{p+q}$ are the acyclic orderings of the strong components of $D - S$ and $D\langle S\rangle$, respectively, and D_{λ_2} is the initial component of D'_2.

Proof: If D is round decomposable and satisfies (a), then we must have $D = R[D_1, D_2, \ldots, D_{p+q}]$, where R is the digraph obtained from D by contracting each D_i into one vertex. This follows from Corollary 2.10.7 and the fact that each of the digraphs $D - S$ and $D - V(D'_2)$ has a round decomposition that agrees with this structure. Now it is easy to see that D does not satisfy (b).

Suppose now that D is not round decomposable. By Lemmas 2.10.12 and 2.10.13, D satisfies (a), so we only have to prove that it also satisfies (b).

If there are no arcs from S to D'_2, then it is easy to see that D has a round decomposition. If there exist components D_{p+i} and D_j with $V(D_j) \subseteq$

$V(D'_2)$, such that there are arcs in both directions between D_{p+i} and D_j, then D satisfies (b). So we can assume that for every pair of sets from the collection $D_1, D_2, \ldots, D_{p+q}$, either there are no arcs between these sets, or one set completely dominates the other. Then, by Corollary 2.10.5, D is round decomposable, with round decomposition $D = R[D_1, D_2, \ldots, D_{p+q}]$ as above, unless we have three subdigraphs $X, Y, Z \in \{D_1, D_2, \ldots, D_{p+q}\}$ such that $X \mapsto Y \mapsto Z \mapsto X$ and there exists a subdigraph $W \in \{D_1, D_2, \ldots, D_{p+q}\} - \{X, Y, Z\}$ such that either $W \mapsto X, Y, Z$ or $X, Y, Z \mapsto W$.

One of the subdigraphs X, Y, Z, say without loss of generality X, is a strong component of $D\langle S\rangle$. If we have $V(Y) \subseteq S$ also, then $V(Z) \subseteq V(D'_2)$ and W is either in $D\langle S\rangle$ or in D'_2 (there are four possible positions for W satisfying that either $W \mapsto X, Y, Z$ or $X, Y, Z \mapsto W$). In each of these cases it is easy to see that D satisfies (b). For example, if W is in $D\langle S\rangle$ and $W \mapsto X, Y, Z$, then any arc from W to Z and from Z to X satisfies the first part of (b). The proof is similar when $V(Y) \subseteq V(D'_3)$. Hence we can assume that $V(Y) \subseteq V(D'_2)$. If $Z = D_p$, then W must be either in $D\langle S\rangle$ and $X, Y, Z \mapsto W$, or $V(W) \subseteq V(D'_2)$ and $W \mapsto X, Y, Z$ (which means that $W = D_i$ and $Y = D_j$ for some $\lambda_2 \leq i < j < p$). In both cases it is easy to see that D satisfies (b). The last case $V(Y), V(Z) \subseteq V(D'_2)$ can be treated similarly. □

We can now state a classification of locally semicomplete digraphs.

Theorem 2.10.15 (Bang-Jensen, Guo, Gutin, Volkmann) *[80] Let D be a connected locally semicomplete digraph. Then exactly one of the following possibilities holds.*

(a) D is round decomposable with a unique round decomposition given by $D = R[D_1, D_2, \ldots, D_\alpha]$, where R is a round local tournament on $\alpha \geq 2$ vertices and D_i is a strong semicomplete digraph for each $i \in [\alpha]$;
(b) D is not round decomposable and not semicomplete and it has the structure as described in Lemma 2.10.14;
(c) D is a semicomplete digraph which is not round decomposable. □

We finish this section with the following useful proposition, whose proof is left as Exercise 2.36.

Proposition 2.10.16 *[80] Let D be a strong non-round decomposable locally semicomplete digraph and let S be a minimal separating set of D such that $D - S$ is not semicomplete. Let $D_1, \ldots, D_p$ be the acyclic ordering of the strong components of $D - S$ and $D_{p+1}, \ldots, D_{p+q}$ be the acyclic ordering of the strong components of $D\langle S\rangle$. Suppose that there is an arc $s \to v$ from S to D'_2 with $s \in V(D_i)$ and $v \in V(D_j)$, then*

$$D_i \cup D_{i+1} \cup \ldots \cup D_{p+q} \mapsto D'_3 \mapsto D_{\lambda_2} \cup \ldots \cup D_j. \qquad \square$$

2.11 Totally Φ-Decomposable Digraphs

Theorem 2.7.5 is a very important starting point for construction of polynomial algorithms for hamiltonian paths and cycles in quasi-transitive digraphs (see Chapter 6) and solving more general problems in this class of digraphs. This theorem shows that quasi-transitive digraphs are totally Φ-decomposable, where Φ is the union of extended semicomplete and transitive digraphs. Since both extended semicomplete digraphs and transitive digraphs are special subclasses of much wider classes of digraphs, it is natural to study totally Φ-decomposable digraphs, where Φ is a much more general class of digraphs than the union of extended semicomplete and transitive digraphs. However, our choice of candidates for the class Φ should be restricted in such a way that we can still construct polynomial algorithms for some important problems such as the hamiltonian cycle problem using properties of digraphs in Φ.

This idea was first used by Bang-Jensen and Gutin [86] to introduce the following three classes of digraphs:

(a) Φ_0 is the union of all semicomplete multipartite digraphs, all connected extended locally semicomplete digraphs and all acyclic digraphs,
(b) Φ_1 is the union of all semicomplete bipartite digraphs, all connected extended locally semicomplete digraphs and all acyclic digraphs, and
(c) Φ_2 is the union of all connected extended locally semicomplete digraphs and all acyclic digraphs.

The aim of this section is to show that totally Φ_i-decomposable digraphs can be recognized in polynomial time for $i = 0, 1, 2$. (If these recognition problems were not polynomial, then the study of the properties of totally Φ_i-decomposable digraphs would be of much less interest.)

A set Φ of digraphs is **hereditary** if $D \in \Phi$ implies that every induced subdigraph of D is in Φ. Observe that every Φ_i, $i = 0, 1, 2$, is a hereditary set.

Lemma 2.11.1 *Let Φ be a hereditary set of digraphs. If a given digraph D is totally Φ-decomposable, then every induced subdigraph D' of D is totally Φ-decomposable. In other words, total Φ-decomposability is a hereditary property.*

Proof: By induction on the number of vertices of D. The claim is obviously true if D has less than 3 vertices.

If $D \in \Phi$, then our claim follows from the fact that Φ is hereditary. So we may assume that $D = R[H_1, \dots, H_r]$, $r \geq 2$, where $R \in \Phi$ and each of $H_1, \dots, H_r$ is totally Φ-decomposable.

Let D' be an induced subdigraph of D. If there is an index i so that $V(D') \subset V(H_i)$, then D' is totally Φ-decomposable by induction. Otherwise, $D' = R'[T_1, \dots, T_{r'}]$, where $r' \geq 2$ and $R' \in \Phi$, is the subdigraph of R

induced by those vertices i of R, whose H_i has a non-empty intersection with $V(D')$ and the T_j's are the corresponding H_i's restricted to the vertices of D'. Observe that $R' \in \Phi$, since Φ is hereditary. Moreover, by induction, each T_j is totally Φ-decomposable, hence so is D'. $\square$

Lemma 2.11.2 *There exists an $O(mn + n^2)$-algorithm for checking if a digraph D with n vertices and m arcs has a decomposition $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where H_i is an arbitrary digraph and the digraph R is either acyclic or semicomplete multipartite or semicomplete bipartite or connected extended locally semicomplete.*

Proof: If D is not connected and $D_1, \ldots, D_c$ are its components, then $D = \overline{K}_c[D_1, \ldots, D_c]$. Hence, in the rest of the proof we may assume that D is connected. We consider the different possibilities for R we are interested in, one by one.

Check whether $\boldsymbol{R}$ can be acyclic: First find the strong components $D_1, \ldots, D_k$ of D. If $k = 1$, then R cannot be acyclic and we can stop verifying that possibility. So suppose $k \geq 2$.

If we find two strong components D_i and D_j such that there is an arc between them but there are non-adjacent vertices $x \in D_i$ and $y \in D_j$, then we replace D_i and D_j by their union. This is justified because D_i and D_j cannot be in different sets H_s and H_t in a possible decomposition. Repeat this step but now check also the possibility for a pair D' and D'' of new 'components' to have arcs between D' and D'' in different directions. In the last case we also replace D' and D'' by their union. Continue this procedure until all remaining sets satisfy that either there is no arc between them, or there are all possible arcs from one to the other. Let $V_1, \ldots, V_r$, $r \geq 1$, denote the distinct vertex sets of the obtained 'components'. If $r = 1$, then we cannot find an acyclic graph as R. Otherwise, $D = R[V_1, \ldots, V_r]$, $r \geq 2$, and we obtain R by taking one vertex from each V_i.

Check whether $\boldsymbol{R}$ can be a semicomplete multipartite digraph: Find the connected components $\overline{G}_1, \ldots, \overline{G}_c$, $c \geq 1$, of the complement of the underlying graph $UG(D)$ of D. If $c = 1$, then R cannot be semicomplete multipartite. So we may assume that $c \geq 2$ below. Let G_j be the subgraph of $UG(D)$ induced by the vertices V_j of the jth component $\overline{G}_j$ of the complement of $UG(D)$. Furthermore, let $G_{j1}, \ldots, G_{jn_j}$, $n_j \geq 1$, be the connected components of G_j. Denote $V_{jk} = V(G_{jk})$.

Starting with the collection $W = \{V_1, \ldots, V_c\}$, we identify two of the sets V_i and V_j if there exist V_{ia} and V_{jb} $a \in [n_i]$, $b \in [n_j]$ such that we have none of the possibilities $V_{ia} \mapsto V_{jb}$, $V_{jb} \mapsto V_{ia}$ or $V_{ia} \rightarrow V_{jb}$ and $V_{jb} \rightarrow V_{ia}$. Clearly the obtained set $V_i \cup V_j$ induces a connected subdigraph of D. Let $Q_1, \ldots, Q_r$ denote the sets obtained, by repeating this process until no more changes occur. If $r = 1$, then R cannot be semicomplete multipartite. Otherwise, R is the semicomplete multipartite digraph obtained by set-contracting each connected component of Q_i into a vertex.

Checking whether R can be a semicomplete bipartite digraph or a connected extended locally semicomplete digraph is left as Exercise 2.39.

It is not difficult to see that, for every R being either acyclic or semicomplete multipartite, the procedures above can be realized as an $O(nm + n^2)$-algorithm. The same complexity is proved for semicomplete bipartite digraphs and extended locally semicomplete digraphs in Exercise 2.39. □

Theorem 2.11.3 *[86] There exists an $O(n^2m+n^3)$-algorithm for checking if a digraph with n vertices and m arcs is totally Φ_i-decomposable for $i = 0, 1, 2$.*

Proof: We describe a recursive algorithm to check Φ_i-decomposability. We have shown in Lemma 2.11.2 how to verify whether $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where R is acyclic, semicomplete multipartite, semicomplete bipartite or connected extended locally semicomplete. Whenever we find an R that could be used, the algorithm checks total Φ_i-decomposability of $H_1, \ldots, H_r$ in recursive calls.

Notice how the algorithm exploits the fact that total Φ_i-decomposability is a hereditary property (see Lemma 2.11.1): if some R is found appropriate, then R can be used, because if D is totally Φ_i-decomposable, then each of $H_1, \ldots, H_r$ (being an induced subdigraph of D) must also be totally Φ_i-decomposable. Since there are $O(n)$ recursive calls, the complexity of the algorithm is $O(n^2m + n^3)$. □

2.12 Planar Digraphs

We now discuss planar (di)graphs, i.e., (di)graphs that can be drawn without crossings between (arcs) edges (except at endpoints). Clearly this property does not depend on the orientation of the arcs and hence we can ignore the orientation below when we give a formal definition. Furthermore, most of the results and definitions in this section are for undirected graphs, but are valid also for planar digraphs as far as their underlying graphs are concerned.

An undirected graph $G = (V, E)$ is **planar** if there exists a mapping f which maps G to $\mathbb{R}^2$ in the following way:

- Each vertex is mapped to a point in $\mathbb{R}^2$ and distinct vertices are mapped to distinct points.
- Each edge $uv \in E$ is mapped to a simple (that is, not self-intersecting) curve C_{uv} from $f(u)$ to $f(v)$ and no two curves corresponding to distinct edges intersect, except possibly at their endpoints.

For algorithmic purposes as well as for arguing about planar graphs, it is inconvenient to allow arbitrary curves in the embeddings of planar graphs. A **polygonal curve** from u to v is a piecewise linear curve consisting of finitely many lines such that the first line starts at u, the last line ends at v and each other line starts at the last point of the previous line. Since we can approximate any simple curve arbitrarily well by a polygonal curve we may assume that the curves used in the embedding are always polygonal curves.

A planar graph G may have many different embeddings in the plane (each embedding corresponds to a mapping f as above). Sometimes we wish to refer to properties of a specific embedding f of G. In this case we say that G is **plane** (that is, already embedded) with planar embedding f. A plane graph G partitions $\mathbb{R}^2$ into a finite number of (topologically) connected regions called **faces**. Precisely one of these faces is unbounded and we call this the **outer face**. It is easy to see that, for any fixed face F of G, we may re-embed G in $\mathbb{R}^2$ in such a way that F becomes the outer face. The boundary of a face F is denoted by $bd(F)$ and we normally describe a face by listing the vertices in clockwise order around the face (for the unbounded face this corresponds to listing the vertices on the boundary in the anti-clockwise order). See Figure 2.19 for an illustration of the definitions.

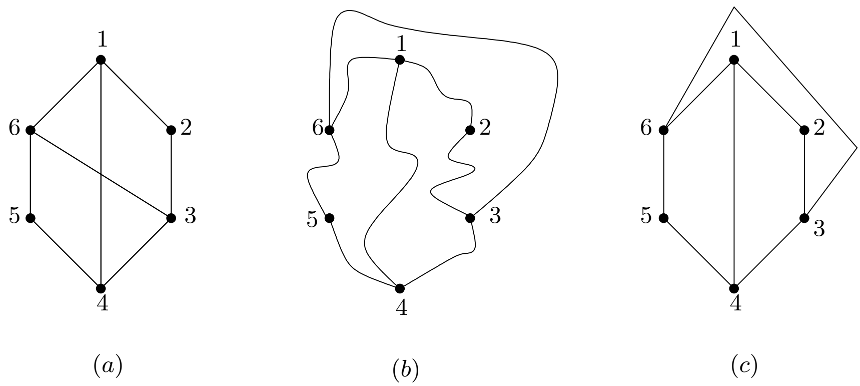


Figure 2.19 (a) shows a non-planar embedding of a graph H; (b) shows a planar embedding of H; (c) shows a planar embedding of H where all curves are polygonal. With respect to the embedding in (c), the faces are 12341, 14561, 16321 and 36543. The outer face is 36543.

Observe that if we add the edge 25 to the graph H in Figure 2.19, then the resulting graph, which is isomorphic to $K_{3,3}$, is no longer planar. In fact, planar graphs have a famous characterization, due to Kuratowski:

Theorem 2.12.1 (Kuratowski's theorem) *[632] A graph has a planar embedding if and only if it does not contain a subdivision*[3] *of* K_5 *or* $K_{3,3}$. □

Based on this it is possible to show that planar graphs (and hence also planar digraphs) can be recognized efficiently. In fact, Hopcroft and Tarjan

[3] A **subdivision** H' of a graph H is any graph that can be obtained from H by replacing each edge by a path all of whose internal vertices have degree 2 in H'.

[534] showed that it can be done in linear time and if the graph is planar, one can find a planar embedding in the same time.

The following relation between the number of vertices, edges and faces in a plane graph, known as **Euler's formula**, is easy to prove by induction on the number of faces.

Theorem 2.12.2 *If G is a connected plane graph on n vertices and m edges, then $n - m + \phi = 2$, where ϕ denotes the number of faces in the embedding on G. In particular, the number of faces is the same in every embedding of G.* □

We leave it to the reader to derive the following easy consequence of Theorem 2.12.2 (see Exercise 2.46):

Corollary 2.12.3 *For every planar graph on $n \geq 3$ vertices and m edges we have $m \leq 3n - 6$.* □

If we allow multiple edges, then we cannot bound the number of edges as we did above. However, for planar digraphs we have the following easy consequence:

Corollary 2.12.4 *No planar digraph on $n \geq 3$ vertices has more than $6n-12$ arcs.* □

We finish this section by a conjecture of Neumann-Lara first posed in 1982 [724] that links planar digraphs with acyclic digraphs.

Conjecture 2.12.5 *The vertices of every planar digraph can be partitioned into two sets such that each set induces an acyclic digraph.*

2.13 Digraphs of Bounded Width

The tree-width is one of the most important parameters in the area of undirected graphs [573]. It is a cornerstone of the Graph Minors Theory, it is used to prove theorems in structural graph theory, and it has many algorithmic applications due to the fact that many $\mathcal{NP}$-hard problems can be solved in linear time when restricted to graphs of bounded tree-width [573]. Naturally, researchers tried to extend the notion of tree-width to digraphs. In particular, Johnson, Robertson, Seymour and Thomas [573] introduced and studied the notion of the directed tree-width, and Berwanger, Dawar, Hunter and Kreutzer [154] and Obdržálek [731] came up with the notion of DAG-width. There are several other directed width parameters, for example, Kelly-with introduced by Hunter and Kreutzer [544].

While the authors of [154, 544, 573, 731] managed to obtain some 'positive' algorithmic results on digraphs of bounded directed tree-width, DAG-width and Kelly-width similar to those on undirected graphs with bounded tree-width, there are several ‘negative’ complexity results obtained by Dankelmann, Gutin and Kim [241] and Kreutzer and Ordyniak [626] indicating that the directed width parameters are of somewhat lesser interest than the tree-width.

In the first subsection of this section we consider digraphs of bounded tree-width and, in the second subsection, we study digraphs in which directed width parameters are bounded.

2.13.1 Digraphs of Bounded Tree-Width

To illustrate the usefulness of tree-width, we will show that one can find, in a linear time, a minimum size kernel[4] in a digraph whose underlying graph is bounded by a constant tree-width. This result allows us to prove that, in a planar digraph D of order n, one can check, in polynomial time, whether D has a kernel of size $O(\log^2 n)$, and if D has such a kernel, then to find one of minimal size.

A non-trivial use of the tree-width is given by Alon, Fomin, Gutin, Krivelevich and Saurabh [21, 22] who proved fixed-parameter tractability of the problem of verifying whether a digraph contains, as a subdigraph, an out-tree with at least k leaves, i.e., vertices of in-degree zero (for the definition of fixed-parameter tractability, see Section 18.4). A refinement of the approach in [21] allowed Bonsma and Dorn [173, 174] to prove fixed-parameter tractability of the problem of verifying whether a digraph has an out-branching with at least k leaves. Another application of tree-width can be found in [472], where Gutin, Razgon and Kim proved that the problem of checking whether a digraph has out-branching with at least k non-leaves is also fixed-parameter tractable.

A **tree decomposition** of an (undirected) graph G is a pair (S, T) where T is a tree whose vertices we will call **nodes** and $S = \{S_i :\ i \in V(T)\}$ is a collection of subsets of $V(G)$ (called **bags**) such that

1. $\bigcup_{i\in V(T)} S_i = V(G)$,
2. for each edge $\{v, w\} \in E(G)$, there is an $i \in V(T)$ such that $v, w \in S_i$, and
3. for each $v \in V(G)$ the set of nodes $\{i :\ v \in S_i\}$ forms a subtree of T.

The **width** of a tree decomposition $(\{S_i :\ i \in V(T)\}, T)$ is defined as the number $\max_{i\in V(T)}\{|S_i| - 1\}$. The **tree-width** of a graph G ($\mathrm{tw}(G)$) is the minimum width over all tree decompositions of G. The **tree-width** of a digraph D ($\mathrm{tw}(D)$) is the tree-width of its underlying graph.

It is not difficult to see that a connected digraph D is of tree-width one if and only if D is a biorientation of a tree (Exercise 2.47). An undirected

[4] A set S of vertices of a digraph D is a **kernel** if S is an independent set and for each $x \in V(D) - S$ there is an out-neighbour in S. For more information on kernels, see Section 3.8.

graph G is called **series-parallel** if there is an ASP digraph D such that $G = UG(D)$. It is well-known (see, e.g., [253] by de Fluiter and Bodlaender) that an undirected graph G has tree-width at most two if and only if each block of G is series-parallel (a **block** of a graph G is a maximal connected subgraph H of G such that $H - x$ is connected for every $x \in V(H)$).

There are several characterizations of undirected graphs of tree-width at most k [601]. We will describe one of the most intuitive such characterizations. A graph G is **chordal**, if every cycle in G of length at least four has a chord, i.e., there is an edge connecting two non-consecutive vertices in the cycle. A **triangulation** of a graph G is a spanning supergraph of G which is a chordal graph.

Theorem 2.13.1 *Let G be a graph with more than k vertices. The graph G is of tree-width at most k if and only if G has a triangulation whose maximum clique has at most $k+1$ vertices.* □

To facilitate our description below we make use of a **nice tree decomposition** (see, e.g., [601] by Kloks). In a nice tree decomposition, we have a binary rooted tree T, i.e., T is a rooted tree such that every node has at most two children. The nodes of T are of four types:

- *An insert node i.* The node i in T has only one child j and there is a vertex $x \in V$ not in S_j such that $S_i = S_j \cup \{x\}$.
- *A forget node i.* The node i in T has only one child j and there is a vertex $x \in V$ not in S_i such that $S_j = S_i \cup \{x\}$.
- *A join node i* has two children p and q. The bags S_i, S_p and S_q are exactly the same.
- *A leaf node i* is simply a leaf of T.

It is not hard to transform a tree decomposition of G into a nice tree decomposition. In fact, the following holds.

Lemma 2.13.2 *[601] Given a tree decomposition of a graph G with n vertices that has width k and $O(n)$ nodes, we can find a nice tree decomposition of G that also has width k and $O(n)$ nodes in time $O(n)$.* □

We will use Lemma 2.13.2 in the following result by Gutin, Kloks, Lee and Yeo [466].

Theorem 2.13.3 *Let D be a digraph of order n. Let the underlying graph G of D have a tree decomposition with $O(n)$ nodes and of width at most t. Then, in $O(n4^t)$ time, we can either find a minimum size kernel in D or determine that D has no kernel.*

Proof: By Lemma 2.13.2, G has a nice tree decomposition with $O(n)$ nodes and of width at most t. Let $(T, \mathcal{S})$ be such a nice tree decomposition of G. Let $S_1, S_2, \ldots, S_r$ be the bags of the tree decomposition (i.e., the nodes of T

are $1, 2, \dots, r$). Let *root* denote the root node of T. Recall that every vertex (and arc) in D lies in at least one of the bags.

Let Y_i denote the union of the bags S_j of the subtree of T with root node i. For every i, consider a partition (K_i, MC_i, DC_i) of S_i (the three sets of a partition are disjoint and every vertex of S_i is in one of the sets). A (K_i, MC_i, DC_i)-**kernel** is an independent set Q in D such that $K_i \subseteq Q \subseteq Y_i$, $(DC_i \cup MC_i) \cap Q = \emptyset$ and every vertex in $Y_i - DC_i$ either lies in Q or has an out-neighbor in Q[5].

The vertices in DC_i may have an out-neighbor in Q, or not. Since $(DC_i \cup MC_i) \cap Q = \emptyset$, every vertex in MC_i has an out-neighbor in Q. We define $\kappa_i(K_i, MC_i, DC_i)$ as the minimal size of a (K_i, MC_i, DC_i)-kernel, if one exists. If it does not exist, then $\kappa_i(K_i, MC_i, DC_i) = \infty$.

If we can compute $\kappa_i(K_i, MC_i, DC_i)$ for all partitions (K_i, MC_i, DC_i) and all i, then

$$\mu = \min\{\kappa_{root}(K, S_{root} - K, \emptyset) : \; K \subseteq S_{root}\} \tag{2.6}$$

gives us the size of a minimum size kernel in D.

Let i be a node of T. We show how to compute, in time $O(4^t)$, all possible $\kappa_i(K_i, MC_i, DC_i)$. In fact we can also compute the actual minimum (K_i, MC_i, DC_i)-kernels, for all possible partitions (K_i, MC_i, DC_i) in $O(4^t)$ time, but we will leave the details of this to the reader. This will imply the desired complexity above as T has $O(n)$ vertices. We consider the cases when i is a leaf, i has one child and i has two children, separately. We assume that if i does have some children, then all κ_i's are known for these children. We will for each step argue that we find the correct values.

Case 1: i is a leaf. There are $O(3^{|S_i|})$ distinct partitions (K_i, MC_i, DC_i), and we can easily find all of these in $O(|S_i|3^{|S_i|})$ time. For each partition (K_i, MC_i, DC_i) we can check whether K_i is an independent set and every vertex in MC_i has an out-neighbor in K_i in time $O(|S_i|^2)$. If the outcomes of both checks are positive, we have $\kappa_i(K_i, MC_i, DC_i) = |K_i|$. Otherwise, we have $\kappa_i(K_i, MC_i, DC_i) = \infty$. This gives us a time complexity of $O(|S_i|3^{|S_i|} + |S_i|^2 3^{|S_i|}) \subseteq O(4^{|S_i|}) \subseteq O(4^t)$ (recall that $|S_i| \le t + 1$).

Case 2: i has one child. Let j be the child of i in T. By the definition of a nice tree decomposition, S_j and S_i are identical, except for one vertex, say x, which lies in either S_i or S_j. We consider the following cases.

If $x \in K_i$, then if x is adjacent to a vertex in K_i, then $\kappa_i(K_i, MC_i, DC_i) = \infty$. Otherwise set $DC_j = DC_i \cup N^-(x)$, $MC_j = MC_i - N^-(x)$ and $K_j = K_i - x$. Clearly $\kappa_i(K_i, MC_i, DC_i) = 1 + \kappa_j(K_j, MC_j, DC_j)$ now holds.

If $x \in MC_i$ and x has no out-neighbor in K_i, then $\kappa_i(K_i, MC_i, DC_i) = \infty$.

If $x \in DC_i$ or $x \in MC_i$ and x has an out-neighbor in K_i, then we have $\kappa_i(K_i, MC_i, DC_i) = \kappa_j(K_i, MC_i - x, DC_i - x)$.

If $x \in S_j$, then we have the following:

[5] MC and DC stand for Must Cover and Don't Care if a vertex from the set has an out-neighbor in the kernel.

$$\kappa_i(K_i, MC_i, DC_i) = \min\{\kappa_j(K_i \cup \{x\}, MC_i, DC_i), \kappa_j(K_i, MC_i \cup \{x\}, DC_i)\}.$$

As all the above cases can be considered in $O(|S_i|)$ time, we get the time complexity $O(|S_i|3^{|S_i|}) = O(4^t)$ for computing κ_i's for all possible partitions.

Case 3: i **has two children.** Let j and l be the two children, and recall that $S_i = S_j = S_l$. It is not difficult to see that $\kappa_i(K_i, MC_i, DC_i)$ is equal to the minimum value of $\kappa_j(K_i, W, MC_i \cup DC_i - W) + \kappa_l(K_i, MC_i - W, DC_i \cup W) - |K_i|$, over all $W \subseteq MC_i$. The above can be done in $O(2^{|MC_i|})$ time and there are $\binom{|S_i|}{m}2^{|S_i|-m}$ partitions (K_i, MC_i, DC_i) with $|MC_i| = m$. Thus, we can compute κ_i's for all possible partitions of S_i in time $O(\sum_{m=0}^{|S_i|} 2^m \binom{|S_i|}{m} 2^{|S_i|-m}) = O(4^t)$.

Since each $\kappa_i(K_i, MC_i, DC_i)$ is computed correctly above, we note that our algorithm will return the correct value of μ in (2.6). If we remember a minimum (K_i, MC_i, DC_i)-kernel for every possible i and partition (K_i, MC_i, DC_i), then our algorithm can in fact return the minimum-sized kernel, and not only its size. Certainly, if $\mu = \infty$, D has no kernel. $\square$

A set S of vertices of an undirected graph G is called **dominating** if for every $x \in V(G) \setminus S$ there is a vertex $s \in S$ adjacent to x. The following result was obtained by Fomin and Thilikos [328].

Theorem 2.13.4 *Let G be a planar graph with n vertices. There is an $O(n^4)$-time algorithm that either constructs a tree decomposition of G with $O(n)$ nodes and of width at most $9.55\sqrt{k}$, or determines that G has no dominating set of size at most k.* $\square$

Observe that every kernel in a digraph D is a dominating set in $UG(D)$. This observation and Theorems 2.13.3 and 2.13.4 imply the following:

Theorem 2.13.5 *[466] Let D be a planar digraph of order n. There is an $O(n2^{19.1\sqrt{k}} + n^4)$-time algorithm that checks whether D has a kernel of size at most k. Moreover, the algorithm finds a minimum size kernel in D, if D has a kernel of size at most k.* $\square$

Theorem 2.13.5 implies that the problem to verify whether a planar graph has a kernel with at most k vertices is fixed-parameter tractable.

Corollary 2.13.6 *[466] Let D be a planar digraph of order n. In polynomial time, one can check whether D has a kernel of size $O(\log^2 n)$, and if D has such a kernel, then find one of minimal size.* $\square$

We conclude this subsection by briefly considering the complexity of checking whether $\mathrm{tw}(G) \leq k$ for a graph G. Unfortunately, the problem is $\mathcal{NP}$-complete, but it is fixed-parameter tractable, and, provided, k is fixed, there is a linear time algorithm for the problem (see [161, 601]).

2.13.2 Digraphs of Bounded Directed Widths

In this subsection, we consider three of the most studied directed width parameters: DAG-widths, directed path-widths and directed tree-width. We will start from the notion of DAG-width rather than that of directed tree-width as the former seems easier to understand than the latter.

A **DAG-decomposition (DAGD)** of a digraph D is a pair (H, χ) where H is an acyclic digraph and $\chi = \{W_h : \ h \in V(H)\}$ is a family of subsets of $V(D)$ satisfying the following three properties: (i) $V(D) = \bigcup_{h \in V(H)} W_h$, (ii) for all $h, h', h'' \in V(H)$, if h' lies on a directed path from h to h'', then $W_h \cap W_{h''} \subseteq W_{h'}$, and (iii) if $(u, v) \in A(D)$, then there exist $h_1, h_2 \in V(H)$ (it is possible that $h_1 = h_2$) such that $u \in W_{h_1}, \ v \in W_{h_2}$ and there is a directed (h_1, h_2)-path in H. The **width** of a DAGD (H, χ) is $\max_{h \in V(H)} |W_h| - 1$. The **DAG-width** of a digraph D ($\mathrm{dagw}(D)$) is the minimum width over all possible DAGDs of D.

A **directed path decomposition (DPD)** is a special case of DAGD when H is a directed path. The **directed path-width** of a digraph D ($\mathrm{dpw}(D)$) is defined as the DAG-width above, but DAGDs are replaced by DPDs.

The following notion of vertex separation allows one to evaluate the directed path-width of a digraph without constructing any DPD. Let D be a digraph and let $\pi = (v_1, v_2, \ldots, v_n)$ be an ordering of $V(D)$. We define $V_j = \{v_i : \ 1 \le j \le i\}$ and $\partial V_j = \{v_i \in V_j : \ (x, v_i) \in A(D)$ for some $x \in V(D) \setminus V_j\}$. With the **vertex separation** of D **with respect to** π given as $\mathrm{vs}_\pi(D) = \max_j |\partial V_j|$, the **vertex separation** of D is defined as $\mathrm{vs}(D) = \min\{\mathrm{vs}_\pi(D) : \ \pi$ is an ordering of $V(D)\}$.

It is well-known that, for undirected graphs, the path-width equals to the vertex separation (see Kirousis and Papadimitriou [597]). We extend this result to digraphs.

Theorem 2.13.7 *For any digraph D,* $\mathrm{vs}(D) = \mathrm{dpw}(D)$.

Proof: Let $\pi = (v_1, v_2, \ldots, v_n)$ be an ordering of $V(D)$ and suppose $\mathrm{vs}_\pi(D) = k$. We will prove that $\mathrm{dpw}(D) \le k$. Set $W_i = \{v_i\} \cup \partial V_{i-1}$ for $i \ge 2$ and $W_1 = \{v_1\}$. We claim that $(12 \ldots n, \chi)$, where $\chi = \{W_1, W_2, \ldots, W_n\}$, is a DPD of width k.

Obviously the property (i) of DPD is satisfied. To check the property (ii), let us choose an arbitrary vertex $v_i \in V(G)$ and see whether the sets W_j containing v_i appear in a row. By the construction of W_j's, the vertex v_i appears in the set W_i and does not appear in any W_j with $j < i$. If there is no backward arc entering v_i, this set is the only one containing v_i and there is nothing to prove. Otherwise let $(v_{i'}, v_i) \in A(D)$ is a backward arc and let i' be the maximum such index. Observe that W_i and W_j for $i < j \le i'$ contain v_i and in fact no other set contains v_i. To check the last property (iii), it is enough to see that both end-vertices of every backward arc $(v_i, v_j) \in A(D)$ are in W_i. It remains to observe that $|W_j| \leq k+1$, which implies that $\mathrm{dpw}(D) \leq k$.

For the converse, let $(12\ldots l, \chi)$, where $\chi = \{W_1, W_2, \ldots, W_l\}$, be a DPD of width k. Without loss of generality we may assume that these sets are all distinct. Let $X_1 = W_1$ and $X_i = W_i \setminus W_{i-1}$ for each $i \geq 2$. Order the vertices of $V(D)$ as follows. We begin with the empty ordering (the 0-th iteration). At the j-th iteration $(1 \leq j \leq l)$ we add a permutation of X_j to the end of the previous iteration ordering. Suppose we have performed all l iterations and obtained an ordering $\pi = (v_1, v_2, \ldots, v_n)$. We will prove that $\mathrm{vs}_\pi(G) \leq k$.

We will prove that $|\partial V_i| \leq k$ for each i. Consider an arbitrary vertex $v_i \in V(D)$ and suppose that v_i was included in π at the j-th iteration, which means $v_i \in W_j$. Notice that $V_i \subseteq W_1 \cup \ldots \cup W_j$. We will first show that $\partial V_i \subseteq W_j$. Consider an arbitrary backward arc (x, y) with $x \in V(D) \setminus V_i$ and $y \in V_i$. Observe that $y \in W_p$ for some $p \leq j$, and if $x \in W_q$ then $q \geq j$. By the property (iii) of DPD, $\{x, y\} \subseteq W_s$ for some $s \geq j$. Thus, by the property (ii) of DPD, $y \in W_j$. Hence, we have shown that $\partial V_i \subseteq W_j$, which implies $|\partial V_i| \leq k+1$. To improve this inequality, we will consider the following two cases:

(a) V_i is a proper subset of $W_1 \cup \ldots \cup W_j$. Then ∂V_j is a proper subset of W_j and $|\partial V_i| \leq k$.

(b) $V_i = W_1 \cup \ldots \cup W_j$. As above we can show that $y \in W_{j'}$ for some $j' > j$. Thus, $y \in W_{j+1}$ and $|\partial V_j| \leq |W_j \cap W_{j+1}| \leq k$. The last inequality holds due to the fact that W_j and W_{j+1} are distinct.

In both cases we conclude that $|\partial V_i| \leq k$, which completes the proof. □

It follows from Theorem 2.13.7 that each directed cycle is of directed path-width 1.

Let Z be a set of vertices of a digraph D. A set $S \subseteq V(D) - Z$ is Z-**normal** if every directed walk that leaves and again enters S must traverse a vertex of Z. For vertices r, r' of an out-tree T we write $r \leq r'$ if there is a path from r to r' or $r = r'$. An **arboreal decomposition** of a digraph D is a triple (R, X, W), where R is an out-tree (not a subdigraph of D), $X = \{X_e : \ e \in A(R)\}$ and $W = \{W_r : \ r \in V(R)\}$ are sets of vertices of D that satisfy two conditions: (1) $\{W_r : r \in V(R)\}$ is a partition of $V(D)$ into nonempty sets, and (2) if for each $e = (r', r'') \in A(R)$ the set $\bigcup\{W_r : \ r \in V(R), r \geq r''\}$ is X_e-normal. The **width** of (R, X, W) is the least integer w such that for all $r \in V(R)$, $|W_r \cup \bigcup_{e \sim r} X_e| \leq w+1$, where $e \sim r$ means that r is head or tail of e. The **directed tree-width** of D, $\mathrm{dtw}(D)$, is the least integer w such that D has an arboreal decomposition of width w.

Now we will study some basic results on the three directed width parameters. The first lemma can be proved using only the definitions above (Exercise 2.48).

Lemma 2.13.8 *Let D be a digraph. For* $\mathrm{d} \in \{\mathrm{dag}, \mathrm{dt}, \mathrm{dp}\}$, *we have* $\mathrm{dw}(D) = 0$ *if and only if D is acyclic.* □

Let D be a digraph. It is immediately follows from the definitions of DAG-width and directed path-width that $\mathrm{dagw}(D) \leq \mathrm{dpw}(D)$. It is easy to show that $\mathrm{dtw}(D) \leq \mathrm{dpw}(D)$ (Exercise 2.49) from the definitions of the two parameters. Berwanger, Dawar, Hunter and Kreutzer [154] proved that $\mathrm{dtw}(D) \leq 3 \cdot \mathrm{dagw}(D) + 1$. Thus, we have the following:

Lemma 2.13.9 *For a digraph D, we have* $\mathrm{dagw}(D) \leq \mathrm{dpw}(D)$, $\mathrm{dtw}(D) \leq \mathrm{dpw}(D)$ *and* $\mathrm{dtw}(D) \leq 3 \cdot \mathrm{dagw}(D) + 1$. □

The last two lemmas imply, in particular, that if $\mathrm{dpw}(D) = 1$ then $\mathrm{dtw}(D) = \mathrm{dagw}(D) = 1$. Thus, for every directed cycle C, we have $\mathrm{dpw}(D) = \mathrm{dtw}(C) = \mathrm{dagw}(C) = 1$. Lemma 2.13.9 has many applications in this book.

Johnson, Robertson, Seymour and Thomas [573] proved that $\mathrm{tw}(G) = \mathrm{dtw}(\overleftrightarrow{G})$ for each undirected graph G and Obdržálek [731] showed that $\mathrm{tw}(G) = \mathrm{dagw}(\overleftrightarrow{G})$ for each undirected graph G. Since the tree-width problem is $\mathcal{NP}$-hard, so are the problems of checking whether $\mathrm{dtw}(D) \leq k$ and $\mathrm{dagw}(D) \leq k$ for a digraph D. However, there are $O(n^{O(k)})$-time algorithms for the two problems [573, 731].

Since $\mathrm{tw}(G) = \mathrm{dtw}(\overleftrightarrow{G})$ and $\mathrm{tw}(G) = \mathrm{dagw}(\overleftrightarrow{G})$ for each undirected graph G, it is easy to prove (Exercise 2.50) that $\mathrm{dtw}(D) \leq \mathrm{tw}(D)$ and $\mathrm{dagw}(D) \leq \mathrm{tw}(D)$ for each digraph D.

2.14 Other Families of Digraphs

This section is devoted to digraphs of three classes: circulant digraphs, arc-locally semicomplete digraphs and intersection digraphs.

2.14.1 Circulant Digraphs

For an integer $n \geq 2$ and a set $S \subseteq \{1, 2, \ldots, n-1\}$, the **circulant digraph** $C_n(S)$ is defined as follows: $V(C_n(S)) = \{1, 2, \ldots, n\}$ and

$$A(C_n(S)) = \{(i, i+j \pmod{n}) : \ 1 \leq i \leq n, j \in S\}.$$

In particular, $C_n(1, 2, \ldots, n-1) = \overleftrightarrow{K}_n$ and $C_n(1) = \vec{C}_n$ (it is customary to omit the set brackets when S is given by a list of its elements). Also, consecutive-1 digraphs introduced at the end of Section 2.5 are circulant digraphs. Circulant digraphs are a special family of Caley digraphs, see, e.g., [568] and are of importance in many applications of graph theory, see, e.g., [269]. Circulant digraphs are of great interest in digraph theory as well, cf. Sections 3.8.1, 6.9 and 15.6. We start from some basic properties of circulant digraphs.

Proposition 2.14.1 *Let $C_n(S)$ be a circulant digraph. Then the following holds:*

(a) $C_n(S)$ *has a 2-cycle if and only if there is a pair* s, t *of elements of* S *such that* $s + t = n$*,*
(b) the converse of $C_n(S)$ *is isomorphic to* $C_n(S)$*,*
(c) $C_n(S)$ *is strong if and only if* $\gcd(n, s_1, s_2, \ldots, s_p) = 1$*, where we have* $\{s_1, s_2, \ldots, s_p\} = S$*.* □

Part (a) is easy to see: if $i \rightarrow j$ and $j \rightarrow i$, then we set $s = i - j$ and $t = j - i$; also, if $s + t = n$, then $(1, 1 + s)$ and $(1 + s, 1 + s + t) = (1 + s, 1)$ are arcs. If n is odd $|S| = (n - 1)/2$, then $C_n(S)$ is a tournament called a **rotational tournament** by Alspach [35]. To prove (b) observe that $C_n(-S)$ is the converse of $C_n(S)$, where $-S = \{-s : s \in S\}$, and that the bijection $f(i) = n - i$ of $[n]$ to itself is an isomorphism of $C_n(S)$ to $C_n(-S)$. It seems Ariyoshi [45] was the first to obtain Part (c); we leave the proof of (c) as an exercise.

In applications it is important to know which circulant digraphs $C_n(S)$ are $|S|$-strong [269] (since $C_n(S)$ is $|S|$-regular, $\kappa(C_n(S)) \leq |S|$ and so $|S|$-strong connectivity is maximal possible for $C_n(S)$). In [269] van Doorn obtained two sufficient conditions:

Theorem 2.14.2 *[269] A circulant digraph* $C_n(S)$ *is* $|S|$*-strong if at least one of the following conditions holds:*

(a) $\gcd(n, s) = 1$ *for each* $s \in S$*,*
(b) $i \in S$ *for each* $i = 1, 2, \ldots, \lceil |S|/2 \rceil$*.* □

2.14.2 Arc-Locally Semicomplete Digraphs

A digraph D is **arc-locally semicomplete** if for every arc xy of D, the following two conditions hold:

(a) if $u \in N^-(x)$, $v \in N^-(y)$ and $u \neq v$, then u and v are adjacent,
(b) if $u \in N^+(x)$, $v \in N^+(y)$ and $u \neq v$, then u and v are adjacent.

This class of digraphs was introduced by Bang-Jensen in [70]. Clearly, every semicomplete or semicomplete bipartite digraph is arc-locally semicomplete. The same holds for extensions of cycles. Bang-Jensen [75] proved that if we restrict ourselves to strong digraphs, the above three classes of digraphs are, in fact, all arc-locally semicomplete digraphs.

Theorem 2.14.3 *[75] A strong arc-locally semicomplete digraph is either semicomplete or semicomplete bipartite or an extension of a cycle.* □

If an arc-locally semicomplete digraph D is non-strong, we do not have a complete picture of how D 'looks like' apart from the case when every vertex of D is on some cycle. In this case, Bang-Jensen [70] showed that D is either semicomplete or semicomplete bipartite. The class of arc-locally semicomplete digraphs was also studied by Galeana-Sánchez [381].

It is natural to define **arc-in-locally (arc-out-locally) semicomplete digraphs** as digraphs satisfying the property (a) (the property (b)) above. To the best of our knowledge, nobody has studied the structure of these two classes of digraphs so far.

2.14.3 Intersection Digraphs

Let U and V be sets and let $\mathcal{F} = \{(S_v, T_v) : S_v, T_v \subseteq U \text{ and } v \in V\}$ be a family of ordered subsets of U (one for each $v \in V$). The **intersection digraph** corresponding to $\mathcal{F}$ is the digraph $D_{\mathcal{F}} = (V, A)$ such that $vw \in A$ if and only if $S_v \cap T_w \neq \emptyset$. The set U is called the **universal set** for $D_{\mathcal{F}}$. The above family of pairs form a **representation** of D. The concept of an intersection digraph is a natural analogue of the notion of an intersection graph and was introduced by Beineke and Zamfirescu [133] and Sen, Das, Roy and West [806]. Since an arc is an ordered pair of vertices, every line digraph $L(D)$ is the intersection digraph of the family $A(D')$, where D' is the converse of D. It follows from the definition of an intersection digraph that every digraph D is the intersection digraph of the family $\{(A^+(v), A^-(v)) : v \in V(D)\}$, where $A^+(v)$ ($A^-(v)$) is the set of arcs leaving v (entering v). Here the universal set is $A(D)$.

Clearly, a digraph can be represented as the intersection digraph of various families of ordered pairs. It is quite natural to ask how large the universal set U has to be. For a digraph D the minimum number of elements in U such that $D = D_{\mathcal{F}}$ for some family $\mathcal{F}$ of ordered pairs of subsets of U is called the **intersection number**, $\mathrm{in}(D)$ of D. Sen, Das, Roy and West [806] prove the following theorem for the intersection number of an arbitrary digraph D. For a digraph $D = (V, A)$, a set $B \subseteq A$ is **one-way** if there is a pair of sets $X, Y \subset V$ (called a **generating pair**) such that $B = (X, Y)_D$, that is, B is the set of arcs from X to Y.

Theorem 2.14.4 *[806] The intersection number of a digraph $D = (V, A)$ equals the minimum number of one-way sets required to cover A.*

Proof: Let $B_1, \ldots, B_k$ be a minimum collection of one-way sets covering A and let $(X_1, Y_1), \ldots, (X_k, Y_k)$ be the corresponding generating pairs. Let $S_v = \{i : v \in X_i\}$, and $T_v = \{i : v \in Y_i\}$. Then $S_v \cap T_w \neq \emptyset$ if and only if $vw \in A$, showing that $\mathrm{in}(D) \leq k$.

Now let U be a universal set of cardinality $u = \mathrm{in}(D)$ such that D has a representation by a set of ordered pairs (S_v, T_v) of subsets of U. We may assume that $U = [u]$. Define u one-way sets covering A as follows: $v \in X_i$ if and only if $i \in S_v$ and $v \in Y_i$ if and only if $i \in T_v$. Then $vw \in A$ if and only if $v \in X_i,\ w \in Y_i$ for some i. Thus, $k \leq \text{in}(D)$. □

Notice that the minimum number of one-way sets required to cover A is studied in Subsection 13.12.1.

A **subtree intersection digraph** is a digraph representable as the intersection digraph of a family of ordered pairs of subtrees in an undirected tree. A **matching diagram digraph** is digraph representable as the intersection digraph of a family of ordered pairs of straight-line segments between two parallel lines. An **interval digraph** is a digraph representable as the intersection digraph of a family of ordered pairs of closed intervals on the real line. Subtree intersection digraphs, matching diagram digraphs and interval digraphs are 'directed' analogues of chordal graphs, permutation graphs and interval graphs, respectively, where subtrees, straight-line segments and real line intervals are also used for representation (see the book [421] by Golumbic). While chordal graphs form a special family of undirected graphs, Harary, Kabell and McMorris showed that every digraph is a subtree intersection digraph.

Proposition 2.14.5 *[499] Every digraph is a subtree intersection digraph.*

Proof: Let $D = (V, A)$ be an arbitrary digraph. Let $G = (U, E)$, $U = V \cup \{x\}$, $E = \{\{x, v\} : \ v \in V\}$, $x \notin V$. Clearly, G is an undirected tree. Setting $S_v = G\langle\{v\}\rangle$ and $T_v = G\langle\{x\} \cup \{w : \ wv \in A\}\rangle$ provides the required representation. □

The following construction by Müller shows that every interval digraph is a matching diagram digraph [708]. Let $\{([a_v, b_v], [c_v, d_v]) : \ v \in V(D)\}$ be a representation of an interval digraph D. To obtain a representation $\{(S_v, T_v) : \ v \in V(D)\}$ of D as a matching diagram digraph we set S_v to be the line segment between points $(a_v, 0)$ and $(b_v, 1)$ in the plane, and T_v to be the line segment connecting the points $(c_v, 1)$ and $(d_v, 0)$.

There are several characterizations of interval digraphs, see, e.g., the papers [793] by Sanyal and Sen and [903] by West. We restrict ourselves to just one of them.

Theorem 2.14.6 *[806] A digraph D is an interval digraph if and only if there exist independent row and column permutations of the adjacency matrix $M(D)$ of D which result in a matrix M' satisfying the following property: the zero entries of M' can be labeled R or C such that every position above and to the right of an R is an R and every position below and to the left of a C is a C.* □

None of the characterizations given in [793, 903] implies a polynomial algorithm to recognize interval digraphs. Müller [708] obtained such an algorithm. A polynomial algorithm is also given in [708] to recognize unit interval digraphs, i.e., interval digraphs that have interval representations, where all intervals are of the same length. Brown, Busch and Lundgren [182] showed that a tournament of order n is an interval digraph if and only if it contains a transitive tournament of order $n-1$ (as a subdigraph).

2.15 Exercises

2.1. **Uniqueness of acyclic orderings.** Prove that an acyclic digraph D has a unique acyclic ordering if and only if D is traceable.

2.2. **Linear time algorithm for finding an acyclic ordering of an acyclic digraph**. Verify that the algorithm given in the proof of Proposition 2.1.3 can be implemented as an $O(n+m)$ algorithm using the adjacency list representation (see Section 18.1).

2.3. Prove that a tournament is transitive if and only if it is acyclic. Hint: apply Theorem 1.5.1.

2.4. Prove Proposition 2.3.1.

2.5. Let D be a semicomplete multipartite digraph such that every vertex of D is on some cycle. Prove that D is unilateral.

2.6. In part (ii) $\Rightarrow$ (i) of Theorem 2.4.1, prove that $\sigma(D) = L(Q)$.

2.7. Derive Corollary 2.4.2 from Theorem 2.4.1 (iii).

2.8. **(−)** Prove Proposition 2.4.3 using Theorem 2.4.1 (i) and (ii).

2.9. Prove the following simple properties of line digraphs:
(i) $L(D) \cong \vec{P}_{n-1}$ if and only if $D \cong \vec{P}_n$;
(ii) $L(D) \cong \vec{C}_n$ if and only if $D \cong \vec{C}_n$.

2.10. Let D be a digraph. Show by induction that $L^k(D)$ is isomorphic to the digraph H, whose vertex set consists of walks of D of length k and a vertex $v_0v_1 \ldots v_k$ dominates the vertex $v_1v_2 \ldots v_kv_{k+1}$ for every $v_{k+1} \in V(D)$ such that $v_kv_{k+1} \in A(D)$.

2.11. Using the results in Exercise 2.9, prove the following elementary properties of iterated line digraphs: Let D be a digraph. Then
(i) $L^k(D)$ is a digraph with no arcs, for some k, if and only if D is acyclic;
(ii) if D has a pair of cycles joined by a path (possibly of length 0), then

$$\lim_{k \longrightarrow \infty} n_k = \infty,$$

where n_k is the order of $L^k(D)$;
(iii) if no pair of cycles of D is joined by a path, then for all sufficiently large values of k, each connected component of $L^k(D)$ has at most one cycle.

2.12. Prove Proposition 2.4.4 by induction on $k \geq 1$.

2.13. Prove Lemma 2.5.1.

2.14. Prove Lemma 2.5.5.

2.15. Prove Lemma 2.5.6.

2.16. Prove Theorem 2.5.7.

2.17. **Upwards embeddings of MVSP digraphs.** Prove that one can embed every MVSP digraph D into the Cartesian plane such that if vertices u, v have coordinates (x_u, y_u) and (x_v, y_v), respectively, and there is a (u, v)-path in D, then $x_u \leq x_v$ and $y_u \leq y_v$. Hint: consider series composition and parallel composition separately.

2.18. Prove Proposition 2.6.2. Hint: use induction on the number of reductions applied for the 'if' part and the number of arcs for the 'only if' part.

2.19. Prove Proposition 2.6.3.

2.20. Prove part (b) of Lemma 2.7.4. Hint: if u and v are in S, then there is a path from u to v in $\overline{UG(S)}$. Similarly, if x and y are in S'. Use these paths (corresponding to sequences of non-adjacent vertices in D) to show that if xu and vy are arcs, then $u = v$ and $x = y$ must hold if D is quasi-transitive.

2.21. **(−)** Construct an infinite family of path-mergeable digraphs, which are not in-path-mergeable.

2.22. **(−)** Show that the following 'claim' is wrong. Let D be a locally in-semicomplete digraph and let D contain internally disjoint paths P_1, P_2 such that P_i is an (x_i, y)-path $(i = 1, 2)$ and $x_1 \neq x_2$. Then x_1 and x_2 are adjacent.

2.23. **Orientations of path-mergeable digraphs.** Prove that every orientation of a path-mergeable digraph is a path-mergeable oriented graph.

2.24. **(+)** Prove Corollary 2.8.2.

2.25. Prove Proposition 2.9.2.

2.26. **Path-mergeable digraphs which are neither locally in-semicomplete nor locally out-semicomplete.** Show by a construction that there exists an infinite class of path-mergeable digraphs, none of which is locally in-semicomplete or locally out-semicomplete. Then extend your construction to arbitrary degrees of vertex-strong connectivity. Hint: consider extensions.

2.27. **(−) Path-mergeable transitive digraphs.** Prove that a transitive digraph $D = (V, A)$ is path-mergeable if and only if for every $x, y \in V$ and every pair xuy, xvy of (x, y)-paths of length 2, either $u{\rightarrow}v$ or $v{\rightarrow}u$ holds.

2.28. Prove Lemma 2.9.3.

2.29. **Orientations of locally in-semicomplete digraphs**. Prove that every orientation of a digraph which is locally in-semicomplete is a locally in-tournament digraph.

2.30. **Strong orientations of strong locally in-semicomplete digraphs**. Prove that every strong locally in-semicomplete digraph on at least three vertices has a strong orientation.

2.31. Prove Lemma 2.10.2.

2.32. Prove Corollary 2.10.7.

2.33. Prove Theorem 2.10.8.

2.34. **(+)** Using Lemma 2.10.13, show that if D is a non-round decomposable locally semicomplete digraph, then the independence number of $UG(D)$ is at most two.

2.35. **(−)** Give an example of a locally semicomplete digraph on 4 vertices with no 2-king.

2.36. Prove Proposition 2.10.16.

2.37. Prove the assertion stated in Exercise 2.34 using Lemma 2.10.14 and Proposition 2.10.16.

2.38. **Extending in-path-mergeability.** Prove that if P, Q are internally disjoint (x, z)- and (y, z)-paths in an extended locally in-semicomplete digraph D and no vertex on $P - z$ is similar to a vertex of $Q - z$, then there is a path R from either x or y to z in D such that $V(R) = V(P) \cup V(Q)$.

2.39. Prove that there exists an $O(mn + n^2)$-algorithm for checking if a digraph D with n vertices and m arcs has a decomposition $D = R[H_1, \ldots, H_r]$, $r \geq 2$, where H_i is an arbitrary digraph and the digraph R is either semicomplete bipartite or connected extended locally semicomplete.

2.40. **(−)** Let D be a connected digraph which is both quasi-transitive and locally semicomplete. Prove that D is semicomplete.

2.41. **(−)** Let D be a connected digraph which is both quasi-transitive and locally in-semicomplete. Prove that the diameter of $UG(D)$ is at most 2.

2.42. **Traceable semicomplete bipartite digraph characterization.** Prove that a semicomplete bipartite digraph B is traceable if and only if it contains a 1-path-cycle factor $\mathcal{F}$. Hint: demonstrate that if $\mathcal{F}$ consists of a path and a cycle only, then B is traceable; use it to establish the desired result (Gutin [445]). (See also Chapter 6.)

2.43. Prove that if a bipartite tournament has a cycle, then it has a 4-cycle.

2.44. Show that every orientation of a quasi-transitive digraph is a quasi-transitive digraph.

2.45. **(−)** Prove that the intersection number $\mathrm{in}(D) \leq n$ for every digraph D of order n. Show that this upper bound is sharp (Sen, Das, Roy and West [806]).

2.46. Prove Corollary 2.12.3. Hint: use that each edge is on the boundary of precisely two faces and that each face has at least three edges.

2.47. Using only the definition of tree-width prove that a connected digraph D is of tree-width one if and only if D is a biorientation of a tree.

2.48. Prove Lemma 2.13.8.

2.49. Prove that $\mathrm{dtw}(D) \leq \mathrm{dpw}(D)$ every digraph D using only the definitions of directed tree-width and directed path-width.

2.50. **(−)** Using the fact that $\mathrm{tw}(G) = \mathrm{dtw}(\overleftrightarrow{G})$ and $\mathrm{tw}(G) = \mathrm{dagw}(\overleftrightarrow{G})$ for each undirected graph G prove that $\mathrm{dtw}(D) \leq \mathrm{tw}(D)$ and $\mathrm{dagw}(D) \leq \mathrm{tw}(D)$ for each digraph D.

2.51. Prove Proposition 2.14.1 (c).