Data Structures and their Planar Graph Layouts

Christopher Auer [1] *Christian Bachmaier* [1] *Franz J. Brandenburg* [1]
Wolfgang Brunner [1] *Andreas Gleißner* [1]

[1]University of Passau, 94030 Passau, Germany

Abstract

In a stack layout, also known as book embedding, the vertices of a graph are placed on a line and an edge is a data item that is pushed on the stack at the left vertex and removed at the right vertex. The LIFO principle of the stack is represented by a rainbow of nesting edges.

We introduce linear cylindric drawings for the representation of the working principles of fundamental data structures including stack, queue, double-stack and deque. The resulting graphs are called stack (queue, double-stack, deque) graphs. We characterize the feasibility of a sequence of insertions and removals by planarity and use the graph classes to compare the relative power of these data structures.

In particular, we show that the deque graphs are the linear cylindric planar graphs and are the subgraphs of the planar graphs with a Hamiltonian path. In comparison, the double-stack graphs are the graphs with a linear layout in the plane and are known as the subgraphs of the planar graphs with a Hamiltonian cycle. Hence, the power of the queue mode of a deque is expressed both by the differences between Hamiltonian path and Hamiltonian cycle and by linear layouts on the cylinder and in the plane. It is also reflected in the dual graph. Linear cylindric drawings provide an intuitive planar representation of the FIFO principle of a queue. We show that a queue graph augmented by a Hamiltonian path has a dual of the same type and that the dual has an Eulerian path. Finally, we study recognition problems.

E-mail addresses: auerc@fim.uni-passau.de (Christopher Auer) bachmaier@fim.uni-passau.de (Christian Bachmaier) brandenb@informatik.uni-passau.de (Franz J. Brandenburg) brunner@fim.uni-passau.de (Wolfgang Brunner) gleissner@fim.uni-passau.de (Andreas Gleißner)

1   Introduction

In his pioneering work on "The Art of Computer Programming" [31] D. E. Knuth studied fundamental data structures including stack, queue, double-stack and double-ended queue (deque). Here, a data structure $\mathcal{D}$ is a (doubly connected) list with insertions and removals of data items only at the ends, called the head and the tail. A sequence of insertions and removals of data items is *feasible* if all data items are inserted and removed at the ends of $\mathcal{D}$. $\mathcal{D}$ is a *stack* if insertions and removals occur at the head, and a *queue* if data items are inserted at the head and removed at the tail. The resulting principles are "last in, first out" (LIFO) for the stack and "first in, first out" (FIFO) for the queue. If each item is removed at the side at which it is inserted, $\mathcal{D}$ is a double-stack and $\mathcal{D}$ is a deque if no restrictions apply.

The behavior of these data structures can be modeled by graphs, where each data item is represented by an edge and the endpoints mark the points of time of its insertion and removal. In a data structure, there is one action at a time which results in graphs with vertices of degree one. For general graphs, actions are grouped at a vertex if the time line of the actions is preserved at each vertex. Our paradigm says that the working principle of $\mathcal{D}$ is satisfied if and only if the graph is drawn planar under specific geometric constraints, which are described by a linear layout.

A *linear layout* of an undirected graph $G = (V, E)$ is a total ordering $\prec$ of the vertices, which are placed left to right on a horizontal line according to $\prec$. It imposes a direction on each edge $e = \{u, v\}$, where we identify e with its directed version $e = (u, v)$ if $u \prec v$. Two edges $e = (u, v)$ and $e' = (u', v')$ with no common endpoints and $u \prec u'$ have three possibilities for the relative positions of their endpoints: e and e' *nest* if $u \prec u' \prec v' \prec v$, *twist* if $u \prec u' \prec v \prec v'$, and are *disjoint* if $u \prec v \prec u' \prec v'$. A graph $G = (V, E)$ has a $\mathcal{D}$ *layout* if there is a linear layout for the vertices and the edges are processed by a data structure $\mathcal{D}$, such that each edge $e = (u, v)$ is inserted to $\mathcal{D}$ at u and removed from $\mathcal{D}$ at v if $u \prec v$ in the linear layout. If the sequence of insertions and removals is feasible, G is called a $\mathcal{D}$ *graph*.

In a stack layout, the edges must obey the LIFO principle and thus either nest or are disjoint; twists are excluded. See Fig. 1(a) for a stack layout (without the dashed edge). The dashed edge twists (with $(1, 3)$) and is invalid. In a queue layout under the FIFO principle the edges twist or are disjoint, whereas nesting edges are excluded. A queue layout (without the dashed edge) is shown in Fig. 1(b). The dashed edge and $(1, 4)$ nest and invalidate the layout. The LIFO principle of a stack is visualized by a rainbow of nesting edges. Such edges do not cross. To the contrary, the FIFO principle of a queue excludes the nesting of edges, and each pair of twisting edges induces a crossing. Thus, it is hard to check the correctness of a queue layout visually and to recognize the FIFO principle. A graph is a *stack graph* if it has a linear layout without twists or, equivalently, a planar drawing respecting the linear layout and with all edges above the line of the vertices, i.e., a 1-page book embedding. Analogously, a *queue graph* has a linear layout without a rainbow.
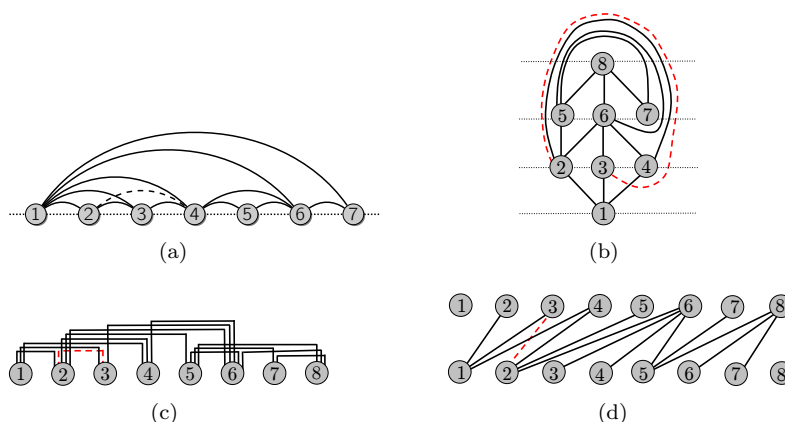
Figure 1: Examples of (a) a stack layout, (b) a queue layout, (c) its arched leveled-planar drawing, and (d) its LC drawing in the fundamental polygon representation. The dashed edges are invalid.

Stack layouts are also known as book embeddings and were studied first by Atneosen [1], Ollman [32] and Kainen [30]. Bernhart and Kainen [9] showed that the stack graphs are exactly the outerplanar graphs. They also characterized the *double-stack graphs*, where the set of edges is partitioned into two disjoint subsets such that no two edges in one subset twist in a given linear layout. The double-stack graphs were characterized as the spanning subgraphs of the planar graphs with a Hamiltonian cycle. The generalization to k-stack layouts or k-page book embeddings are obtained by a partition of the edges into k subsets each of which allows for a stack layout with the same linear ordering of the vertices [16, 18, 38], and to k-stack *subdivisions* or topological book embeddings, where edges can be replaced by paths. Then every planar graph has a 4-stack layout [38] and a 2-stack subdivision [18].

Queue layouts were introduced by Heath, Leighton and Rosenberg [23, 24] and were further studied in [2, 3, 14, 16, 18, 17, 37]. Queue graphs are planar, although a linear drawing of a queue layout contains many crossings. Heath and Rosenberg [24] characterized the queue graphs as the *arched leveled-planar graphs*. These graphs have a planar drawing in which the vertices are placed on horizontal levels. An edge either connects two vertices on adjacent levels or is an arch from the leftmost vertex on a level to vertices on the right side on the same level such that the right vertices are not obstructed by other edges to vertices on the lower level. A linear layout is obtained by ordering the vertices from bottom to top and from left to right on the levels. The arched leveled-planar drawing of the graph from Fig. 1(b) is shown in Fig. 1(c). The dashed edge is invalid as it is an arch that is not incident to the leftmost vertex. A linear cylindric drawing in the fundamental polygon approach is shown in Fig. 1(d). Such layouts make the FIFO-principle of queues clearly visible and are described next.

In this paper, we introduce *linear cylindric drawings* to display stack, queue, double-stack and double-ended queue (deque) layouts, and establish that a drawing is planar if and only if the sequence of insertions and removals of the edges in the data structure is feasible. In Section 3.1, we introduce linear cylindric drawings on the surface of the rolling 3D cylinder. The vertices are placed on a horizontal line, called the *front line*, that is parallel to the cylinder's axis and the edge curves do not cross the front line. Fig. 2(b) shows an example where the front line is dotted. A representation in the plane is obtained by "cutting" the cylinder's surface along the front line and unfolding it (see Fig. 2(c)). The vertices remain on the top side and each vertex has a clone on the bottom side.

In Section 3.3, we apply linear cylindric drawings to deque layouts and prove that the deque graphs are exactly the graphs with a planar linear cylindric drawing. In consequence, all deque graphs are planar, which is unexpected as a deque is an extension of both a double-stack and a queue, the latter of which has many crossings in linear drawings. Linear cylindric planar drawings reflect the working principle of the deque by processing edges that wind around the cylinder as in a queue, whereas edges that enter both endpoints from either above or below the front line are processed as in a stack, as displayed in Fig. 2(b). In fact, the deque can be seen as a double-stack with a transfer of items from one stack to the other. If an item is owned by the stack into which it is inserted, then an item that is processed in queue mode is stolen by the other stack. Hence, the behavior of a deque is not that of a double-ended queue, and the full name is misleading.

Based on these insights, we prove in Section 3.4 that a graph is a deque graph if and only if it is a spanning subgraph of a planar graph with a Hamiltonian path. This result complements the known characterization of double-stack graphs as the spanning subgraphs of planar graphs with a Hamiltonian cycle [9]. Thus, the additional "queue mode" ability of the deque in comparison to the double-stack exactly corresponds to the difference between Hamiltonian paths and cycles in planar graphs, and to planar linear layouts on the surface of a cylinder and in the plane. The difference between double-stack and deque layouts is also visible in the dual graphs, which are investigated in Section 3.5. Note that deques and double-stacks are equivalent in power as working tapes of nondeterministic linear time bounded Turing machines [33].

In Section 4, we apply our findings to queues. In comparison to linear drawings (Fig. 1(b)) and arched leveled-planar drawings (Fig. 1(c)), linear cylindric drawings (Fig. 1(d)) of queue layouts display the FIFO principle, and invalid queue operations are immediately recognizable as crossings. Further, we derive a novel characterization of queue graphs by means of their dual, and show that the dual of a queue graph augmented by a Hamiltonian path is a queue graph augmented by a Hamiltonian path. As a consequence, the dual of a queue graph has an Eulerian path. Recognition problems are studied in Section 5 and we conclude in Section 6 and illustrate new perspectives for linear cylindric drawings.

2   Preliminaries

We consider undirected graphs $G = (V, E)$ with vertices V and edges E. A *path* p in G consists of a sequence of vertices and edges from a vertex u to a vertex v and is denoted by $p = u \rightsquigarrow v$. A path is simple if it contains no vertex twice. A *cycle* C is a path starting and ending at the same vertex, and is simple if all vertices are distinct except at the start and end.

A *linear layout* of G is a total ordering $\prec$ on the set of vertices V. The reflexive closure of $\prec$ is denoted by $\preceq$. A linear layout imposes a direction on each edge $\{u, v\} \in E$ such that $\{u, v\}$ is directed from u to v if $u \prec v$. We identify each edge with its directed version whenever a linear layout is given. If $u \prec v$, then u is a *predecessor* of v and v is a *successor* of u. If there is no vertex w with $u \prec w \prec v$, then u is the *immediate predecessor* of v and v is the *immediate successor* of u. The vertex with no predecessor is called the *first vertex* and the vertex with no successor is called the *last vertex*.

A *drawing* Γ of a graph G maps each vertex $v \in V$ to a position Γ_v in the plane and each edge $e = \{u, v\}$ to an *edge curve* $\Gamma_e : [0, 1] \to \mathbb{R}^2$ which is a simple Jordan arc between its endpoints. The set of points $\{\Gamma_e(t) \mid 0 < t < 1\}$ is the *inner part* of the edge curve Γ_e. We call Γ *planar* if all vertex positions are distinct, no vertex lies on the inner part of an edge curve, and no two edge curves share points except for common endpoints, i. e., they do not cross. A graph is *planar* if it has a planar drawing. A *rotation system* $\mathcal{R}$ defines a cyclic clockwise ordering of the incident edges for each vertex v. We denote the rotation system $\mathcal{R}_v$ at a vertex v by a total ordering $e_0, \ldots, e_{k-1}$ of its incident edges where each edge e_i has a successor edge e_{i+1} and a predecessor edge e_{i-1}, where indices are computed modulo k. Every planar drawing induces a *planar rotation system*. However, there are non-planar rotation systems for planar graphs, e. g., if a K_4 is drawn with a pair of crossing edges. For connected graphs, a planar rotation system is in one-to-one correspondence with a planar embedding which specifies *faces*. Each face f is a connected region which is described by a cycle $C = (v_1, e_1, v_2, e_2, \ldots, v_{k-1}, e_{k-1}, v_k = v_1)$ such that $e_i \in E$ is the successor of $e_{i-1} \in E$ according to the rotation system $\mathcal{R}_{v_i}$. The edges and vertices of C are said to be *incident* to f and C in a *clockwise traversal* of f. The *dual* graph $G^* = (F, E^*)$ has a one-to-one correspondence between the vertices of G^* and the faces of an embedding of G and between the edges of G and G^*. For each edge $e \in E$ there is a *dual edge* $e^* \in E^*$, where $e^* = \{f, g\}$ and f and g are the faces to which e is incident. Note that $f = g$ is possible. For clarity, we always call the elements in V vertices and display them by circular shapes, and we call the elements in F faces and use rectangular shapes. A *cut* is a partition of V into proper subsets $V_C \subsetneq V$ and $\overline{V}_C = V \setminus V_C$. The set of edges $E_C = \{\{u, v\} \in E \mid u \in V_C \wedge v \in \overline{V}_C\}$ that connect vertices in V_C with vertices in $\overline{V}_C$ is called a *cutset*. E_C^* denotes the duals of E_C. The edges of E_C^* constitute a simple cycle in G^* [19, pp. 149]. Conversely, the primal edges of a cutset in G^* constitute a simple cycle in G.

The surface of the 3D *rolling cylinder* is given by $\mathbf{C}_r^3 = \{(x, y, z) \in \mathbb{R}^3 \mid y^2 + z^2 = 1 \wedge -1 \leq x \leq 1\}$. A drawing Γ of a graph on $\mathbf{C}_r^3$ is called *cylindric*. Γ
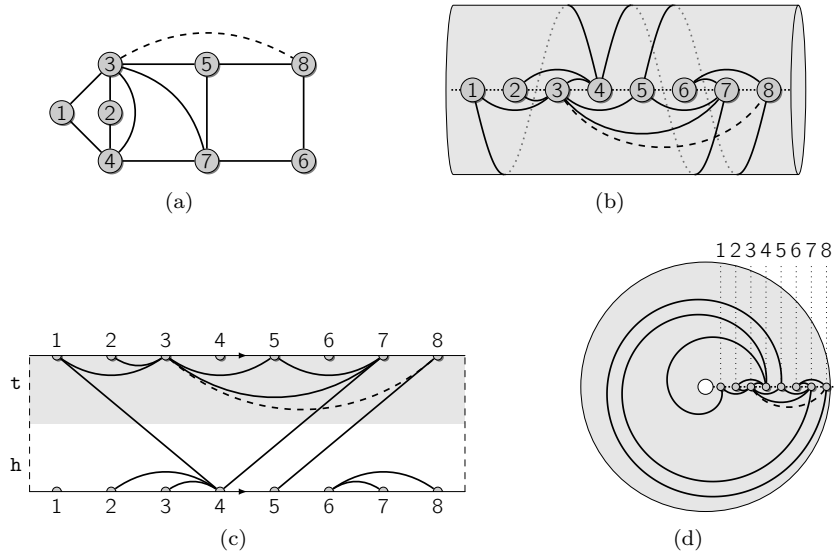
Figure 2: Illustration of graph drawings. (a) a planar graph and a common drawing, (b) an LC drawing as 3D-projection, (c) the fundamental polygon representation and (d) a concentric representation of the LC drawing.

and G are *cylindric planar* if Γ is planar. The standing cylinder is obtained by exchanging the roles of the x and z-coordinates. Note that the cylinder (and the sphere) and the plane are equivalent with respect to planarity, i.e., every planar graph is cylindric planar and vice versa. This does no longer hold true for planar upward drawings of directed planar graphs [4, 10], and for linear layouts, as we shall show in Section 3.1.

3 Deque Graphs and Linear Cylindric Drawings

3.1 Linear Cylindric Drawings

Consider a straight-line segment $h = \{(x, y, z) \in \mathbf{C}_r^3 \mid y = 0 \wedge z = -1\}$ on the cylinder's surfaces which is parallel to the cylinder's axis. We call h the *front line*.

Definition 3.1 *A drawing Γ of a graph $G = (V, E)$ in $\mathbf{C}_r^3$ is* linear cylindric (LC) *if all vertices lie on the front line and no inner part of an edge curve has a point on the front line. The drawing and the graph are called* linear cylindric planar (LC planar) *if the edges do not cross.*

Clearly, every graph has a (not necessarily planar) linear cylindric drawing by arbitrarily placing the vertices on distinct points on the front line and connecting them by edge curves which do not cross the front line. For an illustration,

consider the planar graph from Fig. 2(a) and a possible LC drawing in Fig. 2(b) as a 3D projection. The front line is drawn dotted and the vertices are labeled according to their positions on the front line from left to right. The LC drawing is planar without the dashed edge. Some edges, e.g., $\{1,4\}$, wind around the cylinder. A concentric representation is obtained by mapping the cylinder's surface onto a ring as shown in Fig. 2(d), where the edges that wind around the cylinder are spiral segments. Note that all types of representation are equivalent with respect to planarity since the surfaces have genus 0. Fig. 2(c) shows the fundamental polygon representation of the LC drawing in which the surface of the cylinder corresponds to the rectangle where the top and bottom sides are identified (indicated by the arrows). The front line coincides with the top and bottom of the fundamental polygon. Intuitively, the fundamental polygon representation is obtained from the rolling cylinder by "cutting" the cylinder's surface along the front line and unfolding it.

Proposition 3.1 *An LC planar graph is planar.*

The fundamental polygon representation of LC drawings can also be seen as the drawing of an *extended 2-level planar graph*. In a k-level planar drawing of a graph $G = (V, E)$, the vertices are placed on k horizontal lines and the edges are drawn as vertically strictly monotone curves without crossings. Then the set of vertices partitions into pairwise disjoint sets $V_1, \ldots, V_k$ so that the vertices of V_i are placed on the i-th line. A graph is a k-level planar if it admits such a drawing, where the partition of the set of vertices is given [15, 22, 29].

An *extended k-level drawing* also admits intra-level [8, 26] or track edges [7] between vertices on the same level. The other edges are called inter-level edges. Intra-level edges on level i are drawn as arches above or below the i-th line that do not cross the center line between adjacent levels. The k-level planarity problem, which is solvable in linear time [29], asks whether a graph together with a partition of its set of vertices admits a k-level drawing without edge crossings. In addition, the extended k-level planarity problem has a partition of the set of intra-level edges into the sets of edges above and below each level. It is also solvable in linear time if some requirements are fulfilled [26].

An extended two-level drawing is obtained from a linear cylindric drawing of graph $G = (V, E)$ by taking a clone v' of each vertex v. The vertices of V are placed on the first level and the clones are placed on the second layer such that the vertices and the clones obey the same permutation. If v is the i-th vertex from the left then so is v'. There are edges $\{u, v'\}$ between vertices and clones, which are taken as inter-edges, and edges $\{u, v\}$ and $\{u', v'\}$, which are taken as intra-edges. The recognition problem asks whether there is a planar drawing, i.e., whether there is a single permutation for the vertices on both levels so that there are no crossings. However, this problem is $\mathcal{NP}$-complete, as we shall show in Section 6. In consequence, viewing planar LC drawings as planar two-level drawings may be misleading.

In a planar LC drawing, the vertices are placed at distinct points on the front line which induces a linear layout $\prec$. Further, the edge curve of an edge e
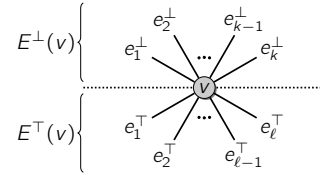
Figure 3: The rotation system at a vertex v defines total orders of the sets $E^\perp(v)$ and $E^\top(v)$.

incident to a vertex v enters v from above or below the front line. For instance, in Fig. 2(c), edge $(2,4)$ enters both of its endpoints from above, whereas edge $(1,4)$ enters 1 from below and 4 from above. This specializes the rotation system. The front line partitions the rotation system of v. Denote by $E^\perp(v) = (e_1^\perp, \ldots, e_k^\perp)$ the edges incident to v that enter v from above, where $E^\perp(v)$ is totally ordered according to the rotation system of v in clockwise direction. Likewise, the edges $E^\top(v) = (e_1^\top, \ldots, e_\ell^\top)$ are the edges that enter v from below and $E^\top(v)$ is totally ordered in counterclockwise direction of v's rotation system. Thus the cw rotation system of v is partitioned into two sequences with a left-to-right order. We call the tuple $\Lambda = (\prec, E^\perp, E^\top)$ an *LC rotation system* and, if it is obtained from a planar LC drawing, we call it *planar*. A rotation system of a vertex v in a planar LC drawing is illustrated in Fig. 3. Note that an LC rotation system is not necessarily planar, since the prescribed order may enforce crossings.

3.2   Deque Layouts

A *deque* is a linear list with two sides, a head $\mathtt{h}$ and a tail $\mathtt{t}$, at which data items are inserted and removed. In our case, data items are edges of a graph. At any time instant, the current state of the deque is given by its content $\mathcal{C} = (e_1, \ldots, e_k)$ $(k \geq 0)$, where e_1 is at the deque's head and e_k at its tail. The deque is empty if $k = 0$. We say $e_i \in \mathcal{C}$ if e_i appears in $\mathcal{C}$. A content $\mathcal{C} = (e_1, e_2, \ldots, e_k)$ induces a total ordering $\ll$ of the edges in $\mathcal{C}$ with $e_1 \ll e_2 \ll \ldots \ll e_k$. If $\mathcal{C} = (e_1, \ldots, e_k)$ $(k \geq 1)$, edge e_1 can be removed at the head and e_k can be removed at the tail which results in $(e_2, \ldots, e_k)$ and $(e_1, \ldots, e_{k-1})$, respectively. These and only these removals are feasible. Inserting an edge e at the head or tail transforms a deque's content $\mathcal{C} = (e_1, \ldots, e_k)$ to $\mathcal{C} = (e, e_1, \ldots, e_k)$ or $\mathcal{C} = (e_1, \ldots, e_k, e)$, respectively.

In a graph layout by the deque, the vertices are visited according to a linear layout $\prec$ and the edges are processed at their endpoints. We start with an empty deque at the first vertex. At each vertex v, all incoming edges from predecessors must be removed from the deque and all outgoing edges to successors must be inserted into the deque. Each edge is processed at the head or the tail. For each vertex v, let $E^\mathtt{h}(v) = (e_1^\mathtt{h}, \ldots, e_k^\mathtt{h})$ and $E^\mathtt{t}(v) = (e_1^\mathtt{t}, \ldots, e_\ell^\mathtt{t})$ be a partition of v's incident edges into two sets, both of which are totally ordered. The edges in $E^\mathtt{h}(v)$ $(E^\mathtt{t}(v))$ are processed at the head (tail) of the deque in order. That

is, at vertex v, starting with $e_1^{\mathtt{h}}$, all edges in $E^{\mathtt{h}}(v)$ are inserted and removed at the head in order, depending on whether they are incoming or outgoing. Likewise, all edges in $E^{\mathtt{t}}(v)$ are processed at the tail in order. We call the tuple $\Sigma = (\prec, E^{\mathtt{h}}, E^{\mathtt{t}})$ a *deque schedule*. A deque schedule is called a *deque layout* if all operations are feasible. A graph is a *deque graph* if it has a deque layout. Note that an edge can always be inserted at either side of a deque. However, at the removal it must be at the head or tail, as otherwise another edge is blocking its way and the deque cannot operate properly. At each vertex v, we assume that all edges in $E^{\mathtt{h}}(v)$ are processed before the edges in $E^{\mathtt{t}}(v)$, since operations at opposite sides do not interfere. An edge e is called *stack edge* if it is inserted and removed at the same side and *queue edge* otherwise.

Suppose that two edges e and e' are processed at the head such that e is removed and e' is inserted. Then inserting e' first would imply that $e' \ll e$, which makes it impossible to remove e at the head. Hence, there is a local ordering of incoming and outgoing edges.

Proposition 3.2 *Let $\Sigma = (\prec, E^{\mathtt{h}}, E^{\mathtt{t}})$ be a deque layout. For each vertex v, $E^{\mathtt{h}}(v) = (e_1^{\mathtt{h}}, \ldots, e_k^{\mathtt{h}})$ can be divided into $E_{\mathrm{Pred}}^{\mathtt{h}}(v) = (e_1^{\mathtt{h}}, \ldots, e_i^{\mathtt{h}})$ and $E_{\mathrm{Succ}}^{\mathtt{h}}(v) = (e_{i+1}^{\mathtt{h}}, \ldots, e_k^{\mathtt{h}})$ for some $0 \le i \le k$ such that all edges in $E_{\mathrm{Pred}}^{\mathtt{h}}(v)$ are removed at the head, and all edges in $E_{\mathrm{Succ}}^{\mathtt{h}}(v)$ are inserted at the head. If $i = 0$, then $E_{\mathrm{Pred}}^{\mathtt{h}}(v) = \emptyset$ and, if $i = k$, then $E_{\mathrm{Succ}}^{\mathtt{h}}(v) = \emptyset$. The same holds true for $E^{\mathtt{t}}(v)$.*

Processing the edges of a deque graph G according to a deque schedule Σ is described by a *trajectory* $\tau(\Sigma)$. The trajectory maps each vertex to two points in the plane and each edge to a polyline.

Let $v_1 \prec v_2 \prec \ldots \prec v_n$ be the linear layout of G and for each vertex v_i, $i = 1, \ldots, n$, let $E^{\mathtt{h}}(v_i) = E_{\mathrm{Pred}}^{\mathtt{h}}(v_i) \cdot E_{\mathrm{Succ}}^{\mathtt{h}}(v_i)$ and $E^{\mathtt{t}}(v_i) = E_{\mathrm{Pred}}^{\mathtt{t}}(v_i) \cdot E_{\mathrm{Succ}}^{\mathtt{t}}(v_i)$. Let $s_0 = 0$ and let s_i denote the size of the deque after v_i. Then $s_{i+1} = s_i - |E_{\mathrm{Pred}}^{\mathtt{h}}(v_i)| + |E_{\mathrm{Succ}}^{\mathtt{h}}(v_i)| - |E_{\mathrm{Pred}}^{\mathtt{t}}(v_i)| + |E_{\mathrm{Succ}}^{\mathtt{t}}(v_i)|$ for $i = 0, \ldots, n-1$, where $s_i \ge 0$ and $s_n = 0$ and $|L|$ denotes the length of list L. The deque is empty at the start and at the end.

The trajectory $\tau(\Sigma)$ maps each vertex v_i to the points $(i, 0)$ and $(i, \max\{s_{i-1}, s_i\})$, called the head and tail positions. It can be seen as a two-level drawing. Each processed edge at v_i is drawn as a straight line to a point on a small circle of radius 0.25 around the positions of v_i. The edges of $E_{\mathrm{Succ}}^{\mathtt{h}}(v_i)$ are in the first quadrant and those of $E_{\mathrm{Pred}}^{\mathtt{h}}(v_i)$ are in the second quadrant of the head position $(i, 0)$, and those of $E_{\mathrm{Pred}}^{\mathtt{t}}(v_i)$ and $E_{\mathrm{Succ}}^{\mathtt{t}}(v_i)$ are in the third and fourth quadrants of the tail position, respectively, as illustrated in Fig. 3. If $C(v_i) = (e_1, \ldots, e_k)$ is the content of the deque after v_i, then there is a bend point $P(v_i, e_j) = (i + 0.5, j - 0.5)$ for each edge e_j. If edge e is inserted at the head of the deque at v_i, then draw a straight line from the endpoint of e on the circle around the head point of v_i to $P(v_i, e)$, and if e is removed at v_i, then draw a straight line from $P(v_{i-1}, e)$ to the endpoint of e on the circle, and similarly if e is processed at the tail. If edge e remains in the deque, then draw a straight line from $P(v_{i-1}, e)$ to $P(v_i, e)$. Note that the position of e in the deque may change due to insertions and removals of edges at v_i, and e may be the h-th edge from the head of the deque before v_i
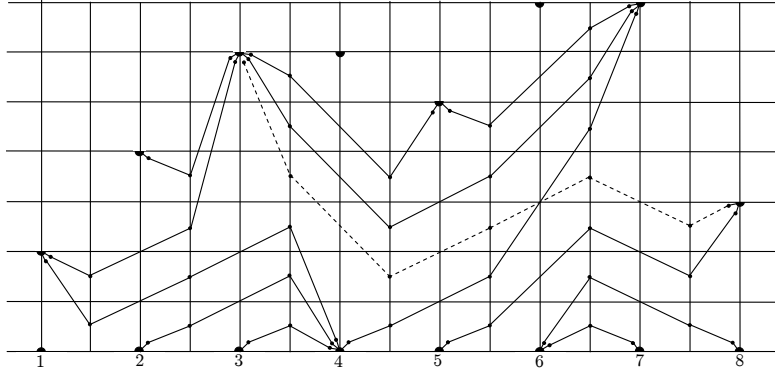
Figure 4: The trajectory of the deque schedule Σ from Fig. 2(b).

and the j-th edge after v_i. Clearly, j can be computed from h and the deque schedule Σ.

The trajectory of an edge e is a polyline that begins at the head or tail position of vertex v_i where e is inserted and ends at vertex v_j where e is removed. Intermediately, it first passes a point on the circle around v_i in the first or fourth quadrant, then it goes through points $P(v_p, e)$ for $p = i, \ldots, j-1$ and finally it enters the circle around v_j in the second or third quadrant. The trajectory of the Σ from Fig. 2(b) is shown in Fig. 4, including the illegal edge $(3, 8)$. The fundamental polygon representation in Fig. 2(c) is a smoothed version with all tail positions moved to the upper boundary.

3.3   Deque Graphs and Planar Linear Cylindric Graphs

Our first main result describes the power of a deque in terms of linear cylindric planar graphs, and conversely, shows that linear cylindric planar graphs can be processed in a deque.

Theorem 1 *A graph is a deque graph if and only if it is LC planar.*

We use an LC rotation system $\Lambda = (\prec, E^\perp, E^\top)$ for an LC drawing of a graph G and describe a graph layout in the deque by a deque schedule $\Sigma = (\prec, E^{\mathsf{h}}, E^{\mathsf{t}})$. Both are the same objects with the same linear ordering of the vertices and a partition of the incident edges at each vertex. Even more, the equivalence "planar $\Leftrightarrow$ feasible" holds, and this implies Theorem 1.

There is a partition for planar LC rotation systems similar to the one for deque layouts given in Proposition 3.2, as Fig. 5 illustrates.

Lemma 1 *Let $\Lambda = (\prec, E^\perp, E^\top)$ be a planar LC rotation system. For each vertex v, the total ordering of the edges in $E^\perp(v) = (e_1^\perp, \ldots, e_k^\perp)$ can be divided into $E_{\mathrm{Pred}}^\perp(v) = (e_1^\perp, \ldots, e_i^\perp)$ and $E_{\mathrm{Succ}}^\perp(v) = (e_{i+1}^\perp, \ldots, e_k^\perp)$ for some $0 \le i \le k$ such that all edges in $E_{\mathrm{Pred}}^\perp(v)$ are incident to predecessors of v, and all edges in*
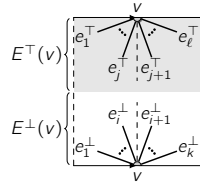
Figure 5: Structure of a planar LC rotation system.

Figure 6: Illustration to the proof of Lemma 1: A crossing is inevitable if an edge $e_j^\perp$ to a predecessor comes after an edge $e_i^\perp$ to a successor, where edge $e_j^\perp$ enters the bottom line (a) from above or (b) from below.

$E_{\mathrm{Succ}}^\perp(v)$ are incident to successors. If $i = 0$, then $E_{\mathrm{Pred}}^\perp(v) = \emptyset$ and, if $i = k$, then $E_{\mathrm{Succ}}^\perp(v) = \emptyset$. The same holds true for $E^\top(v)$.

Proof: For contradiction, suppose that there is a vertex v and edges $e_i^\perp$ and $e_j^\perp$ in $E^\perp(v) = (e_1^\perp, \ldots, e_k^\perp)$ such that $i < j$ and $e_i^\perp$ is incident to a successor w of v and $e_j^\perp$ is incident to a predecessor u of v. The situation is depicted in Fig. 6(a). As the inner part of the edge curve of $e_j^\perp$ has no point in common with the front line, the edge curve of $e_j^\perp$ and the front line enclose a region R (shaded in Fig. 6(a)). Edge $e_i^\perp$ enters v from above and has no point in common with the front line except for its endpoints, and thus crosses the boundary of R. The other endpoint w of $e_i^\perp$ lies outside R as if w would lie within, an inner point of $e_j^\perp$'s edge curve would lie on the front line. Consequently, the inner part of $e_i^\perp$'s edge curve has points inside and outside R which inevitably leads to a crossing by Jordan's curve theorem; a contradiction to the planarity of the LC rotation system. The proof for $E^\top(v)$ is analogous.

Similarly, if $e_j^\perp$ enters u from below, it encloses a region R together with the front line as illustrated in Fig. 6(b) and the proof is analogous. $\square$

Fig. 5 shows how the rotation system in a planar LC rotation system is split according to Lemma 1.

Lemma 2 *Let $\Sigma = (\prec, E^{\mathrm{h}}, E^{\mathrm{t}})$ be a deque schedule and let $\Lambda = (\prec, E^\perp, E^\top)$ be an LC rotation system with $\Sigma = \Lambda$. Σ is a deque layout if and only if Λ is planar.*

Proof: $\Rightarrow$: Consider the trajectory $\tau(\Sigma)$ of a deque layout Σ. The polylines of two edges e and e' do not cross, since once they are inserted, their ordering in the
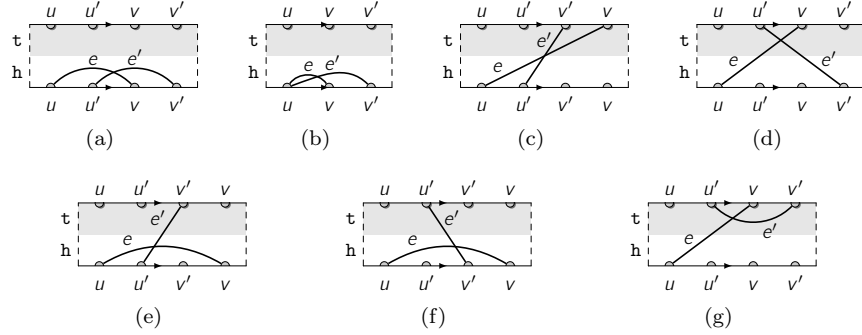
Figure 7: Cases obtained in the proof of Lemma 2. (a) Both edges are stack edges processed at the head. (b) Both edges are stack edges incident to u and processed at the head. (c) Both edges are queue edges inserted at the same side. (d) Both edges are queue edges inserted at different sides. (e) e is a stack and e' a queue edge inserted at the head. (f) e is a stack and e' a queue edge inserted at the tail. (g) e is a queue and e' a stack edge.

deque is unchanged by insertions and removals of edges at the vertices. Hence, $\tau(\Sigma)$ is planar. Now, transform $\tau(\Sigma)$ into an LC planar drawing by moving the tail position of vertex v_i to (i, s), where s is the maximum size of the deque and then map all points to the surface of a cylinder by a transformation $(x, y) \mapsto (\frac{x}{n}, \frac{2\pi}{s} \cos y, \frac{2\pi}{s} \sin y)$. Obviously, both transformations preserve planarity.

$\Leftarrow$: In the following, we assume that Λ is a planar LC rotation system but, towards a contradiction, Σ is no deque layout. As insertions to the deque never cause trouble, there must be an edge which cannot be removed from the deque because another edge is blocking its way. Let $e = (u, v)$ and $e' = (u', v')$ with $u \prec v$ and $u' \prec v'$ be two such edges. We assume that e is inserted to the deque before e' such that $u \preceq u' \prec v$ by Lemma 1. We distinguish five cases according to the side of the insertion and removal of the edges.

Case 1: Both e and e' are stack edges. We assume that both edges are inserted and removed at the head. As e is inserted before e', we get $e' \ll e$. Removing e is not possible if and only if e is removed before e' and, thus, $v \preceq v'$, where for $v = v'$ edge e comes before e' in $E^{\mathrm{h}}(v)$. If $v \prec v'$, then $u \preceq u' \prec v \prec v'$ and we obtain a twist (Fig. 7(a)) and there is an inevitable crossing between e and e'. If $v = v'$ and e comes before e' in $E^{\mathrm{h}}(v)$, we also obtain an inevitable crossing as shown in Fig. 7(b). Hence, Λ was not planar.

Case 2: Both e and e' are queue edges inserted at the same side. If both edges are inserted at the head, then $e' \ll e$ and we have no deque layout if and only if e' has to be removed before e. Hence, $u \preceq u' \prec v' \preceq v$ and e nests e'. We obtain the situation in Fig. 7(c); again a contradiction to the planarity of Λ. The reasoning is analogous if e and e' are inserted at the tail.

Case 3: Both e and e' are queue edges inserted at different sides. In the LC drawing, e and e' always cross as shown in Fig. 7(d).

Case 4: e is a stack and e' a queue edge. We assume that e is inserted at the head and distinguish two cases: either e' is inserted at the head or at the tail. In the first case, $e' \ll e$ and neither can e be removed from the head nor e' from the tail. We obtain the situation in Fig. 7(e) and Λ cannot be planar. If e' is inserted at the tail, then $e \ll e'$. Now, removing e from the head is always possible. However, removing e' before e is not possible. This is the case if and only if $v' \prec v$, or $v' = v$ and e' comes before e in $E^{\mathtt{h}}(v)$. For the LC drawing, we obtain the situation in Fig. 7(f) and an inevitable crossing.

Case 5: e is a queue and e' a stack edge. By swapping the role of e with e', and $\mathtt{h}$ with $\mathtt{t}$, and reversing the linear layout, the proof of this case is analogous to the previous one.

Altogether, we have obtained a contradiction and can conclude that Σ is a deque layout. $\qquad\square$

For an illustration, consider Fig. 2(c), where we ignore the dashed edge for the moment. From the planar LC drawing, we obtain the planar LC rotation system $\Lambda = (\prec, E^{\perp}, E^{\top})$, which is used as a deque schedule, and Fig. 4 shows the obtained trajectory. The ordering in which the vertices are processed for the deque layout is equal to the ordering of the vertices on the front line of the LC drawing. The head and tail correspond to the regions above and below the front line, respectively. If $e = (v, w)$ is an edge with $e \in E^{\perp}(v)$, e.g., edge $(2, 4)$ in Fig. 2(c) in the LC layout, then $e \in E^{\mathtt{h}}(v)$ in the deque layout. Likewise, any edge in $E^{\top}(v)$ is processed at the tail of the deque. An edge is a queue edge if and only if it winds around the cylinder and has its endpoints on opposite sides of the fundamental polygon. In Fig. 2(c), stack edges remain inside the regions of the head or the tail, which is shaded. The content of the deque is given by a vertical scan line from bottom to top. The dashed edge $(3, 8)$ causes an error at vertex 7, since the content is $((6, 7), (6, 8), (5, 8), (4, 7), (3, 8), (3, 7), (5, 7))$ and $(3, 8)$ blocks the removal of $(4, 7)$. Furthermore, at each vertex the incoming edges from predecessors must be removed before the outgoing edges to successors.

From Theorem 1 and Proposition 3.1 we obtain:

Corollary 3.1 *Deque graphs are planar.*

3.4   Deque Graphs and Hamiltonian Paths

The double-stack graphs are the planar sub-Hamiltonian graphs [9], where Hamiltonian means the class of graphs with a Hamiltonian cycle. Here we extend this characterization to deque graphs and Hamiltonian paths.

Theorem 2 *A graph is a deque graph if and only if it is a spanning subgraph of a planar graph with a Hamiltonian path.*

Proof: Consider an LC drawing of G according to the deque schedule and an LC rotation system $\Sigma = (\prec, E^{\mathtt{h}}, E^{\mathtt{t}})$, which is planar by Lemma 2. Augment G by the edges $E' = \{(u, v) \notin E \mid v \text{ is the immediate successor of } u \text{ according to } \prec\}$. Then $G' = (V, E \cup E')$ has a Hamiltonian path given by $\prec$. The edges of E' can
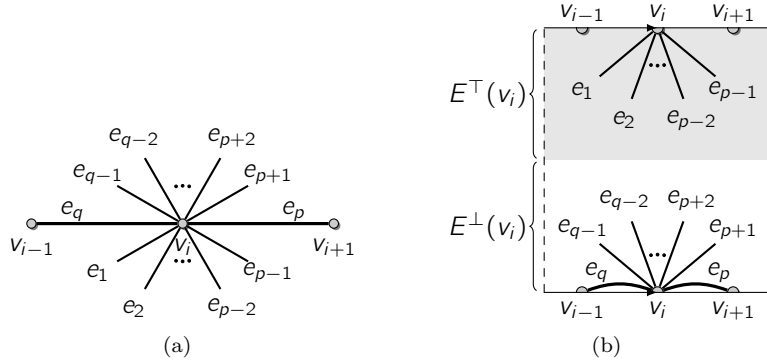
Figure 8: (a) Rotation system of an inner vertex v_i of a graph with a Hamiltonian path and (b) the obtained LC rotation system.

be drawn close to the front line of the cylinder and do not cross other edges, so that G' is planar.

For the converse, let $G = (V, E)$ be a planar graph with a Hamiltonian path $P = (v_1, \ldots, v_n)$ and endowed with a planar clockwise rotation system $\mathcal{R}$. Let $n \geq 2$ as otherwise G consists of a single vertex and is a deque graph. We construct a planar LC rotation system $\Lambda_P = (\prec, E^\perp, E^\top)$ from G's rotation system and the Hamiltonian path P, where P plays the role of a guideline and splits the rotation system at each vertex into an upper and a lower part. The linear layout $\prec$ is the ordering of the vertices on P. Let $\mathcal{R}_{v_i} = (e_1, \ldots, e_{p-1}, e_p, e_{p+1}, \ldots, e_q)$ be the rotation system of v_i. For each inner vertex v_i with $1 < i < n$ on P, let e_p be the edge to the immediate successor v_{i+1} of v_i and e_q the edge to the immediate predecessor v_{i-1} of v_i on the Hamiltonian path. Fig. 8(a) illustrates the rotation system of v_i, where the edges e_p and e_q are drawn bold. Let $E^\perp(v_i) = (e_q, e_{q-1}, \ldots, e_{p+1}, e_p)$ and $E^\top(v_i) = (e_1, e_2, \ldots, e_{p-2}, e_{p-1})$. Note the reverse ordering in $E^\perp(v_i)$. The so obtained LC rotation system is illustrated in Fig. 8(b). For the extreme vertices v_1 and v_n on P, we define the LC rotation system of v_1 (v_n) such that all edges enter v_1 (v_n) from above. If e_p is the first edge on P to v_2 and e_q is the last edge from v_{n-1} to v_n, then $E^\perp(v_1) = (e_{p-1}, e_{p-2}, \ldots, e_2, e_1, e_p)$ such that $E^\top(v_1)$ is empty, and $E^\perp(v_n) = (e_q, e_{q-1}, \ldots, e_2, e_1)$ and $E^\top(v_n)$ is empty.

It remains to show that there is a planar LC drawing of G that respects the constructed LC rotation system $\Lambda_P = (\prec, E^\perp, E^\top)$. By the construction of Λ_P, the Hamiltonian path is aligned with the front line and planarity ensures that no edge curve has as a point in common with the front line. We now show that Λ_P is planar. For contradiction, assume that every such LC drawing has at least one crossing. Let e and e' be two crossing edges, where $e = \{v_i, v_j\}$. There is a cycle C formed by the edge e and the path from v_i to v_j which is part of the Hamiltonian path. Circle C encloses a region R such that at least one inner point of the edge curve of e' lies within R and one inner point lies outside R.

Recall that G is endowed with a planar rotation system $\mathcal{R}$ and, hence, it has a planar drawing Γ which induces $\mathcal{R}$. By construction, $\mathcal{R}$ is the rotation system induced by the LC drawing Γ^P. Thus, cycle C encloses a region $\overline{R}$ in Γ such that at least one inner point of the edge curve of e' must lie within $\overline{R}$ and one inner point must lie outside, which causes a crossing in Γ by Jordan's curve theorem. This contradicts the planarity of Γ and, hence, Λ_P is planar. $\square$

The constructions from the second part of the proof of Theorem 2 lead to the following generalization. Let G be a graph endowed with a not necessarily planar rotation system $\mathcal{R}$ and assume that G contains a Hamiltonian path P. Let $\Lambda_P = (\prec, E^\perp, E^\top)$ be the LC rotation system obtained from $\mathcal{R}$ and P, such that P splits the rotation systems at the vertices and defines a deque schedule $\Sigma_P = \Lambda_P$. Λ_P and Σ_P are called *P-induced*. From Theorems 1 and 2 we obtain:

Corollary 3.2 *Let G be a graph endowed with a rotation system and assume that G contains a Hamiltonian path P. Further, let Λ_P and Σ_P be the P-induced LC rotation system and P-induced deque schedule, respectively. The following statements are equivalent:*

(i) The rotation system of G is planar.

(ii) Λ_P is a planar LC rotation system.

(iii) Σ_P is a deque layout.

The relative power of double-stacks and deques can now be compared from the viewpoint of linear graph layouts. A double-stack is a specialization of a deque. The surplus of a deque is the "queue mode" with insertions and removals at opposite sides. What does this really mean? How large is the gap between a double-stack and a deque? The queue mode of a deque is reflected by the queue edges. These wind around the cylinder, and so they preserve the FIFO structure. This property is captured by upward planar drawings of directed graphs. Upward means that the edge curves must be monotone in y-direction. Here, the plane is strictly weaker than the sphere which is equivalent to the standing cylinder [4, 10, 13, 21]. The difference between upward planar graphs in the plane and on the sphere lies in an (s, t)-edge from the lowest to the highest vertex, which prevents edges from winding around the cylinder.

On the other hand, the ability to process queue edges in addition to stack edges exactly corresponds to the difference between Hamiltonian paths and Hamiltonian cycles in planar graphs. Moreover, we shall show in Section 3.5 that the missing edge closing the Hamiltonian cycle crosses all queue edges.

3.5 Duals of Deque Graphs

The difference between double-stack and deque graphs is also reflected in the dual graphs. Consider the maximal planar graph depicted in Fig. 9(a) which contains the Hamiltonian cycle $(1, \ldots, 8, 1)$, where the edges of the Hamiltonian
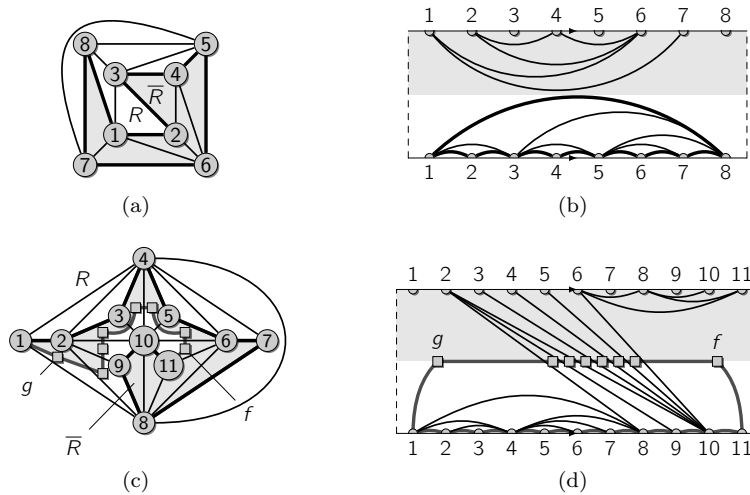
Figure 9: (a) A maximal planar graph with a Hamiltonian cycle and (b) the obtained double-stack layout. (c) The Goldner-Harary graph with a Hamiltonian path and (d) its deque layout.

cycle are drawn bold. The Hamiltonian cycle defines a closed curve C that divides the plane into regions R and $\overline{R} := \mathbb{R}^2 \setminus R$ as shown in Fig. 9(a), where $\overline{R}$ is shaded. Now, the inner part of each edge not on the Hamiltonian cycle either lies completely in R or in $\overline{R}$. By identifying R with the head and $\overline{R}$ with the tail of a deque, we obtain the double-stack layout displayed in Fig. 9(b), where the linear layout is obtained by splitting the Hamiltonian cycle between vertices 8 and 1.

We can apply the same line of arguments to deque layouts, however, with an intriguing twist. Fig. 9(c) shows the Goldner-Harary graph G_{GH} [20] which is the smallest maximal planar graph with no Hamiltonian cycle. The edges drawn bold belong to the Hamiltonian path $(1, \ldots, 11)$. Therefore, G_{GH} is a deque but no double-stack graph. We use the same idea as before to construct the deque layout. Consider the open curve defined by the Hamiltonian path in Fig. 9(c). We close the curve by proceeding in the dual graph. First, connect vertex 11 with face f to which 11 is incident. The choice of face f is arbitrary as long as vertex 11 is incident to f. Then, we find a path in the dual graph from face f to face g, where vertex 1 is incident to g, such that we never use any dual edge of the Hamiltonian path. Finally, connect g to 1 and close the curve. Thereby, we obtain a simple closed curve C that partitions the plane into regions R and $\overline{R}$ (shaded). Again, identify R with the head and $\overline{R}$ with the tail. The linear layout is the ordering of the vertices on the Hamiltonian path. The corresponding LC drawing is shown in Fig. 9(d). As with the double-stack graph from before, we have edge curves within either R or $\overline{R}$ which are stack edges, e. g., edge $(1, 4)$. Consider the edges that cross C. For instance, edge $(3, 10)$ starts within $\overline{R}$ and

ends within R which means that it is inserted at the tail and removed at the head of the deque. In general, all queue edges cross C and they are exactly the primal of the edges on the path from f to g in the dual. We comprise these observations in the following theorem which we use later to characterize queue graphs:

Theorem 3 *Let $G = (V, E)$ be a graph with a planar rotation system and a Hamiltonian path $P = (v_1, \ldots, v_n)$ consisting of edges E_P. Let $G^* = (F, E^*)$ be the dual of G and E_P^* the dual edges of E_P. Further, let $f_1, f_p \in F$ be two faces such that v_1 is incident to f_p and v_n is incident to f_1. Then:*

(i) There is a simple path $Q^* = f_1 \rightsquigarrow f_p$ in $G_{\overline{P}}^* = (F, E^* \setminus E_P^*)$.

(ii) Let E_Q^* be the set of edges traversed by Q^* and let E_Q be the primal edges of E_Q^*. G has a deque layout $\Sigma = (\prec, E^{\mathtt{h}}, E^{\mathtt{t}})$ such that $\prec$ is the ordering of the vertices on P, and an edge $e \in E$ is a queue edge if and only if $e \in E_Q$.

Proof: (i): Suppose for contradiction that there is no path $f_1 \rightsquigarrow f_p$ in $G_{\overline{P}}^*$. Since G has a Hamiltonian path, G is connected and so is its dual G^*. Hence, there is a path from f_1 to f_p in G^* but removing the dual edges of the Hamiltonian path disconnects f_1 from f_p. Denote by F_C the set of faces f for which there is a path $f_1 \rightsquigarrow f$ in $G_{\overline{P}}^*$. By construction, $f_1 \in F_C$ and $f_p \in \overline{F}_C = F \setminus F_C$. Then, F_C and $\overline{F}_C$ is a cut with cutset $E_C^* \subseteq E^*$. Let $e^* = \{f, g\}$ be an edge from E_C^* with $f \in F_C$ and $g \in \overline{F}_C$. The dual edge e^* must be in E_P^*, as otherwise there would be a path from f_1 to g. Hence, $E_C^* \subseteq E_P^*$ and $E_C \subseteq E_P$, where E_C are the primal edges of E_C^*. As E_C^* is a cutset, E_C forms a cycle in G. However, then the Hamiltonian path contains a cycle which is a contradiction. Therefore, there is a simple path from f_1 to f_p.

(ii): It was shown by Tutte [34] that a planar graph $G = (V, E)$ and its dual $G = (F, E^*)$ can be drawn simultaneously such that each of G and G^* is drawn planar, all faces are placed within the regions to which they correspond, and each primal edge e crosses its dual e^* exactly once. Let Γ be such a simultaneous drawing of G and G^*. The situation is illustrated in Fig. 10(a). Denote by Γ_P the set of points in Γ of the simple curve defined by the Hamiltonian path P in G (drawn bold in Fig. 10(a)). By (i), there is a simple path $Q^* = f_1 \rightsquigarrow f_p$ in G^* (shaded line between f_1 and f_p). Γ_{Q^*} denotes the set of points in Γ that belong to Q^*. Since Q^* contains no edge of E_P^*, Γ_P and Γ_{Q^*} are two disjoint simple curves. We connect these two curves as follows: Let Γ_{v_1} be the position of v_1 and Γ_{f_p} be the position of f_p. Since v_1 is incident to f_p, there is a simple curve from v_1 to f_p that shares no point with Γ except for its endpoints (dashed line between v_1 and f_p). Likewise, there is a simple curve from v_n to f_1 (dashed line between v_n and f_1). Altogether, we obtain the simple and closed curve C that partitions $\mathbb{R}^2$ into regions R and $\overline{R}$ (shaded).

Let $\Lambda = (\prec, E^\perp, E^\top)$ be the LC rotation system induced by the Hamiltonian path. Recall that the Hamiltonian path P splits the rotation system of each vertex v into the edges that enter v from above and from below, and the rotation
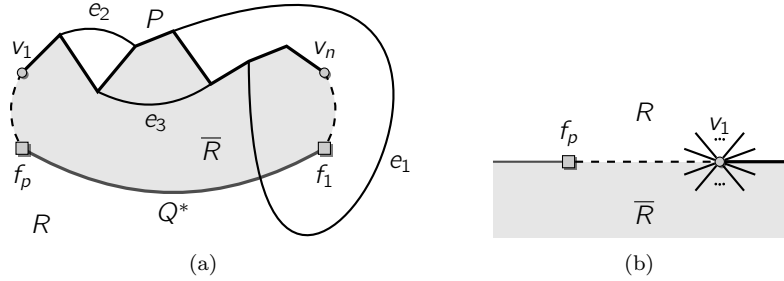
Figure 10: Situation obtained in the proof of Theorem 3. (a) Connecting the endpoints of the Hamiltonian path P and Q^* encloses the region $\overline{\overline{R}}$. (b) The rotation system of v_1 is split by the Hamiltonian path and by the curve connecting v_1 and f_1.

systems of v_1 and v_n can be split arbitrarily. In the construction of C, we have connected v_1 with f_p by a curve which can be seen as an edge curve. Together with the edge to vertex v_2, the rotation system of v_1 is split as shown in Fig. 10(b), where the curve between v_1 and f_p is dashed. The edges that enter v_1 within R $(\overline{R})$, enter v_1 from above (below) in the LC rotation system Λ. Likewise, the rotation system of v_n is split. The so obtained LC rotation system is planar, as shown in the proof of Theorem 2, and the corresponding deque schedule $\Sigma = (\prec, E^{\mathtt{h}}, E^{\mathtt{t}})$ is a deque layout by Corollary 3.2.

Now, let e^* be an edge on Q^* and let $e = \{u, v\}$ be its primal. By construction, the edge curve of e crosses C exactly once, e. g., e_1 in Fig. 10(a). This implies that e either enters u within R and v within $\overline{R}$ or vice versa. Hence, e enters u from above and v from below or vice versa. In both cases, e is a queue edge in Σ. Further, let e be an edge such that the inner part of its edge curve lies completely within R, e. g., edge e_2. Then, e enters both its endpoints from above and is, thus, a stack edge inserted at the head in Σ. Likewise, an edge that lies within $\overline{R}$ is a stack edge inserted at the tail in Σ. This proves (ii).   □

In conclusion, the additional power of a deque over a double-stack is to connect the endpoints of the Hamiltonian path via the dual, which in turn produces queue edges. Conversely, if a graph can be augmented to a planar graph with a Hamiltonian path such that its endpoints are incident to the same face, we can connect these endpoints in the dual without traversing any dual edges. Hence, we need no queue edges and the graph is a double-stack graph.

4   Queue Graphs

Our results on deque graphs have relevant implications on queue graphs. A queue is a restricted deque, where all edges are queue edges inserted at the head and removed at the tail. With this specialization we have *queue schedules* $\Sigma = (\prec, E^{\mathtt{h}}, E^{\mathtt{t}})$, *queue layouts* and *queue graphs* and we can conclude:
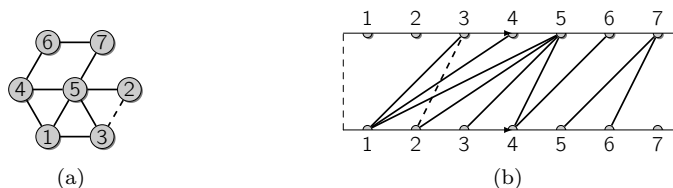
Figure 11: (a) A planar graph and (b) its queue layout.

Corollary 4.1 *A queue schedule Σ is a queue layout if and only if the LC rotation system Λ with $\Lambda = \Sigma$ is planar.*

Corollary 4.2 *A graph is a queue graph if and only if it has a planar LC drawing such that all edges $e = (u, v)$ enter u from above and v from below the front line, i.e., they wind around the cylinder.*

Now there are three ways to display queue layouts. First, in the common way, where each twist generates a crossing [16, 18, 24], which is bad from the viewpoint of graph drawing (Fig. 1(b)). Second, the arch leveled drawings from [24] are planar, however, they do not reflect the FIFO principle (Fig. 1(c)). Also, it is quite cumbersome to construct such a drawing, and to check an infeasibility. Why is the dashed edge in Fig. 1(c) illegal? If we keep this edge, which others must be removed? Finally, our linear cylindric drawings (see Fig. 11(b)) are planar, the FIFO principle is made transparent by a scan line and illegal edges are directly recognizable.

The restriction to queue edges has a strong impact on the dual of queue graphs. The addition of the edges between the vertices that are consecutive in the linear layout or on a Hamilton path results in a multi-graph, whose distinguished property is that the dual is of the same type. The added edges are not necessarily queue edges; they could be used as stack edges in a deque, and are taken for the Hamilton path, which implies that the original queue edges are all crossed in the dual graph when connecting the endpoints of the Hamiltonian path.

Given a linear layout $\prec$ and a graph $G = (V, E_Q)$, the $\prec$-*augmentation* $G_\prec = (V, E_Q \,\dot\cup\, E_P)$ of G is a multi-graph with disjoint sets of edges E_Q and $E_P = \{\{u, v\} \mid u, v \in V$ and v is the immediate successor of u in $\prec\}$. We specialize Theorem 3 for queue graphs.

Lemma 3 *Let $G = (V, E_Q)$ be a queue graph with queue layout $\Sigma = (\prec, E^{\mathrm{h}}, E^{\mathrm{t}})$ and $\prec$-augmentation $G_\prec = (V, E_Q \,\dot\cup\, E_P)$. Let v_1 and v_n be the first and last vertex of the Hamiltonian path with edges E_P. Then $G_\prec$ has an embedding with dual $G_\prec^* = (F_\prec, E_Q^* \,\dot\cup\, E_P^*)$ such that there exist faces $f_1, f_p \in F_\prec$, where v_n is incident to f_1 and v_1 is incident to f_p, and there is a simple path $f_1 \rightsquigarrow f_p$ in $G_\prec^* \setminus E_P^*$ that contains all edges E_Q^*.*

Proof: Let $\Sigma_\prec$ be the $\prec$-augmentation of Σ, where $\Sigma_\prec$ is a deque schedule of $G_\prec$ that equals Σ for all edges in E_Q and all edges in E_P are stack edges processed at

the head. Note that $\Sigma_\prec$ is not necessarily a queue layout. Let $\Lambda_\prec = (\prec, E^\perp, E^\top)$ be the LC rotation system with $\Lambda_\prec = \Sigma_\prec$. Then $\Lambda_\prec$ is planar by Lemma 2, and we obtain an embedding of $G_\prec$ with dual graph $G_\prec^* = (F_\prec, E_Q^* \dot\cup E_P^*)$. Further, $G_\prec$ contains the Hamiltonian path E_P with first vertex v_1 and last vertex v_n. Now, we apply Theorem 3 and choose f_1 and f_p so that all edges in E_Q that are incident to v_1 enter v_1 from above and all edges in E_Q incident to v_n enter v_n from below. Hence, there is a simple path $Q^* = f_1 \rightsquigarrow f_p$ in $G_\prec^* \setminus E_P^*$ that contains each edge in E_Q^*. □

Given a $\prec$-augmentation $G_\prec = (V, E_Q \dot\cup E_P)$ of a queue graph, we say that an embedding of $G_\prec$ is an *augmented queue embedding* (with Hamiltonian path E_P and queue edges E_Q) if it fulfills the properties of Lemma 3. This augmented queue embedding can readily be constructed following the proof of Lemma 3.

Corollary 4.3 *A graph $G = (V, E_Q)$ is a queue graph if and only if it is a spanning subgraph of a $\prec$-augmentation $G_\prec = (V, E_Q \dot\cup E_P)$ such that $G_\prec$ has an augmented queue embedding.*

Proof: The only if direction follows from Lemma 3. For the if-direction, note that the augmented queue embedding of $G_\prec$ and the Hamiltonian path induce a deque layout $\Sigma_\prec = (\prec, E^h, E^t)$ where all edges in E_P are stack edges and all edges in E_Q are queue edges. Removing all edges E_P from $G_\prec$ yields G, which is then a queue graph. □

Figs. 12(a) and 12(b) show the augmented queue embedding of $G_\prec = (V, E_Q \dot\cup E_P)$ (solid) and its dual $G_\prec^* = (F, E_Q^* \dot\cup E_P^*)$ (dashed) of the queue graph in Fig. 11(b) (without the dashed edge). Note that all edges of E_P enter their endpoints from above, whereas all edges in E_Q change sides. $G_\prec$ is a multigraph as it contains the edge $(6, 7)$ twice. Interestingly, the dual has the same type as the primal: The duals of E_Q, denoted by E_Q^*, are a Hamiltonian path $Q^* = f_1, \ldots, f_{10}$ in $G_\prec^*$. Let $\prec^*$ be the linear layout of $G_\prec^*$ in order of Q^*, e.g., $f_i \prec^* f_j \Leftrightarrow i < j$ for all $1 \le i, j \le 10$ in Figs. 12(a) and 12(b). The duals of E_P, denoted by E_P^*, behave like queue edges: For instance, the dual of $(4, 5) \in E_P$ is edge (f_3, f_8) which enters f_3 from below and f_8 from above. Thus, the embedding of $G_\prec^*$ is an augmented queue embedding with Hamiltonian path E_Q^* and queue edges E_P^*. Put differently, Hamiltonian path edges and queue edges switch roles when going from the primal to the dual. In fact, this characterizes augmented queue embeddings:

Theorem 4 *Let $G = (V, E_Q \dot\cup E_P)$ be an embedded graph such that E_P is a Hamiltonian path, and let $G^* = (F, E_Q^* \dot\cup E_P^*)$ be the dual of G. The following are equivalent:*

(i) *The embedding of G is an augmented queue embedding, where E_P is the Hamiltonian path and E_Q are the queue edges.*

(ii) *The embedding of G^* is an augmented queue embedding, where E_Q^* is the Hamiltonian path and E_P^* are the queue edges.*
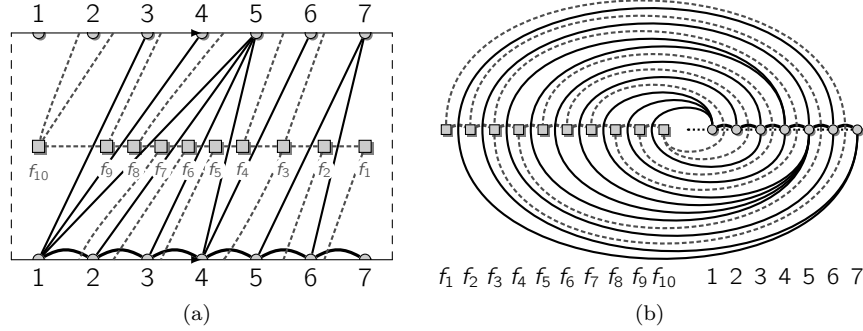
Figure 12: The augmented queue embedding and its dual (dashed) of the queue graph in Fig. 11(b), (a) in fundamental polygon representation and (b) in concentric representation.

Proof: (i) $\Rightarrow$ (ii): Let $v_1, \ldots, v_n$ be the Hamiltonian path with edges E_P and let $\prec$ be the respective linear layout. As the embedding of G is an augmented queue embedding, there exist two faces f_1 and f_p such that v_n is incident to f_1, v_1 is incident to f_p, and there is a simple path $Q^* = f_1 \leadsto f_p$ in $G^* \setminus E_P^*$ that contains all edges E_Q^* by Lemma 3.

First, we show that Q^* is a Hamiltonian path of G^*. Every face $f \in F$ has at least one incident edge of E_Q^*, otherwise there is a face f with incident edges E_f^* such that $E_f^* \subseteq E_P^*$. Then $E_f^* \neq \emptyset$, since the dual is connected. As E_f^* is a cutset, the primal edges of E_f^* form a simple cycle in G that consists of Hamiltonian path edges, which is a contradiction. As Q^* visits each face at least once, it contains all edges in E_Q^*. Further, Q^* is simple and, thus, is a Hamiltonian path of G^* from f_1 to f_p.

Let $\Lambda^* = (\prec^*, E^{*\perp}, E^{*\top})$ be the LC rotation system of G^* as induced by the Hamiltonian path Q^*. We can assume that all edges incident to f_1 enter f_1 from above and all edges incident to f_p enter f_p from below. Let $\Sigma^* = (\prec, E^{*\mathbf{h}}, E^{*\mathbf{t}})$ be the deque schedule with $\Sigma^* = \Lambda^*$. Due to planarity and Theorem 3, Λ^* is planar and Σ^* is a deque layout. It remains to show that all edges in E_P^* are queue edges in Σ^* that are inserted at the head and removed at the tail.

As in the proof of Theorem 3, we assume that we are given a simultaneous drawing of G and G^*, which are both drawn planar according to the given LC rotation systems, all faces are placed within the regions to which they correspond, and each primal edge e crosses its dual e^* exactly once. For every edge $e \in E_P \dot\cup E_Q$, let $l(e)$ and $r(e)$ denote the endpoints of e with $v_1 \preceq l(e) \prec r(e) \preceq v_n$. Then $r(e)$ is the immediate successor of $l(e)$ on the Hamiltonian path E_P if $e \in E_P$, and if $e \in E_Q$, then e is inserted at $l(e)$ and removed at $r(e)$ in the queue layout of G. Analogously, for every edge $e^* \in E_P^* \dot\cup E_Q^*$, $l(e^*)$ and $r(e^*)$ are the faces to which e is incident and $f_1 \preceq^* l(e^*) \prec^* r(e^*) \preceq^* f_p$. By assumption, all edges incident to f_1 enter f_1 from above and, hence, are inserted at the head in Σ^*. Likewise, all edges incident to f_p are removed at the tail.

Let $e^* \in E_P^*$ be an edge such that $f_1 \prec^* l(e^*) \prec^* r(e^*) \prec^* f_p$. In the remainder of the proof, we show that e^* enters $l(e^*)$ from above and $r(e^*)$ from below the Hamiltonian path E_Q^*. The situation is depicted in Fig. 13, where again the dual edges are drawn dashed for clarity. The dual edge e^* must lie in the shaded region, which forms a "tube" and guarantees that e^* enters its endpoints from the correct sides. We start by defining the elements that form the border of the "tube".

As $l(e^*) \prec^* r(e^*)$ there is an edge $e_1^* \in E_Q^*$ on the Hamiltonian path of G^* such that $l(e^*) \preceq^* l(e_1^*) \prec^* r(e_1^*) \preceq^* r(e^*)$. Its primal e_1 is a queue edge and enters $l(e_1)$ from above and $r(e_1)$ from below by our assumption. For the endpoints of e_1, we show that:

$$v_1 \preceq l(e_1) \preceq l(e) \prec r(e) \preceq r(e_1) \preceq v_n. \tag{1}$$

First, e_1 crosses its dual e_1^* between $l(e^*)$ and $r(e^*)$ on the Hamiltonian path of G^* since $l(e^*) \preceq^* l(e_1^*) \prec^* r(e_1^*) \preceq^* r(e^*)$. Second, e_1 enters $l(e_1)$ from above and $r(e_1)$ from below by assumption. Hence, if $l(e) \prec l(e_1)$ or $r(e_1) \prec r(e)$, then e_1 would cross e^* which is not possible as e_1 is not the primal of e^*. Note that this crossing exists independently of the sides from which e^* enters its endpoints. Hence, Eq. (1) follows.
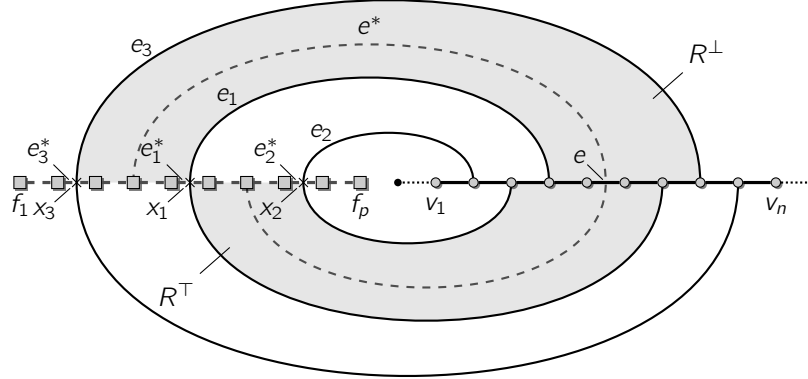

Figure 13: Situation obtained in the proof of Theorem 4.

Further, as $r(e^*) \prec^* f_p$, there is an edge $e_2^* \in E_Q^*$ such that $r(e^*) \preceq^* l(e_2^*) \prec^* r(e_2^*) \preceq^* f_p$. For the endpoints of the primal e_2 of e_2^*, we show that:

$$v_1 \preceq l(e_2) \preceq l(e_1) \quad \text{and} \quad l(e_2) \prec r(e_2) \preceq l(e). \tag{2}$$

The edge curves of e_1 and the edges on the Hamiltonian path between $l(e_1)$ and $r(e_1)$ enclose a region R in the drawing, where, w.l.o.g., f_p is situated within R. By definition, the endpoints of e_2^* lie between $r(e^*)$ and f_p on the Hamiltonian path of G^*. Thus, the position of the crossing of e_2 with e_2^* is between $r(e^*)$ and f_p and, in particular, between the crossing of e_1^* with e_1 and the position

of f_p. Recall that vertex v_1 is incident to f_p. For these reasons and due to the planarity of the drawing of G, the inner part of the edge curve of e_2 lies completely within R. Further, e_2 enters $l(e_2)$ from above and $r(e_2)$ from below and, hence, $v_1 \preceq l(e_2) \preceq l(e_1)$ and $l(e_2) \prec r(e_2) \preceq r(e_1)$. It remains to show that $r(e_2) \preceq l(e)$. For this, consider the crossing of e^* with e. If $l(e) \prec r(e_2)$, then e_2 would cross e^* which is not possible as e^* is not the dual of e_2. Hence, Eq. (2) follows.

By symmetric arguments, there is an edge $e_3^* \in E_Q^*$ such that $f_1 \preceq^* l(e_3^*) \prec^* r(e_3^*) \preceq^* l(e^*)$, where the primal e_3 of e_3^* has the properties:

$$r(e) \preceq l(e_3) \prec r(e_3) \quad \text{and} \quad r(e_1) \preceq r(e_3) \preceq v_n \,. \tag{3}$$

Consider the bounded region $R^\perp$ enclosed by the curve that consists of the following elements of the drawing (shaded and above the front line in Fig. 13):

- The edge curves on the Hamiltonian path E_P of G from $l(e_1)$ to $l(e_3)$, where $l(e_1) \preceq l(e) \prec r(e) \preceq l(e_3)$ by Eqs. (1) and (3).

- The part of the edge curve of e_3 from $l(e_3)$ to the crossing point x_3 of e_3 with dual edge e_3^*.

- The edge curves of the Hamiltonian path E_Q^* from x_3 to the crossing point x_1 between e_1 and e_1^*.

- The part of the edge curve of e_1 from x_1 to $l(e_1)$, which closes the curve.

Additionally, we define the bounded region $R^\top$ which is enclosed by the following elements of the drawing (shaded and below the front line in Fig. 13):

- The edge curves on the Hamiltonian path E_P of G from $r(e_2)$ to $r(e_1)$, where $r(e_2) \preceq l(e) \prec r(e) \preceq r(e_1)$ by Eqs. (1) and (2).

- The part of the edge curve of e_1 from $r(e_1)$ to the crossing point x_1 of e_1 with dual edge e_1^*.

- The edge curves of the Hamiltonian path E_Q^* of G^* from x_1 to the crossing point x_2 of e_2 with e_2^*.

- The part of the edge curve of e_2 from x_2 to $r(e_2)$, which closes the curve.

Note that due to the properties of e_1, e_2, and e_3, both curves we have just constructed are non-selfintersecting and, thus, $R^\perp$ and $R^\top$ are well defined. They force the edge curve of e^* to enter its endpoints from the correct sides. First note that the crossing point of e with e^* lies on the boundaries of both $R^\perp$ and $R^\top$ as $l(e_1) \preceq l(e) \prec r(e) \preceq l(e_3)$ and $r(e_2) \prec l(e) \prec r(e) \preceq r(e_1)$, respectively. Moreover, $l(e^*)$ is on the boundary of $R^\perp$ and $r(e^*)$ is on the boundary of $R^\top$. Hence, all inner points of the edge curve of e^* are within $R^\perp \cup R^\top$ as otherwise e^* cannot cross its primal e. Consequently, e^* enters $l(e^*)$ from within $R^\perp$ and, by the construction of $R^\perp$, e^* must enter $l(e^*)$ from above. Similarly, e^* enters $r(e^*)$ from within $R^\top$ and, thus, from below. We can
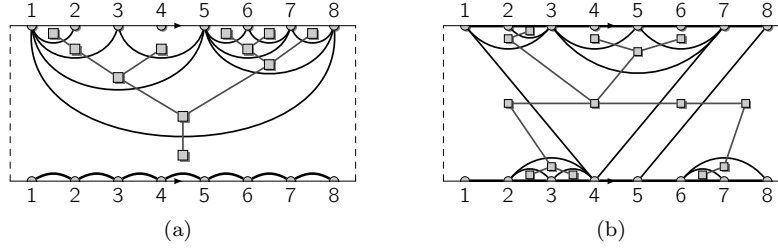
Figure 14: The dual (a) of an augmented stack graph and (b) an augmented deque graph.

conclude that e^* is a queue edge in the deque schedule Σ^* inserted at the head and removed at the tail.

For an edge e^* with endpoint f_1, an analogous reasoning shows that e^* is removed at the tail, and if e^* has endpoint f_p, it is inserted at the head.

(ii) $\Rightarrow$ (i): The proof is similar with swapped roles of primal and dual.   □

From Corollary 4.3 and Theorem 4, we obtain the following characterization of queue graphs:

Corollary 4.4 *For a graph $G = (V, E_Q)$, the following are equivalent.*

  (i) *G is a queue graph.*

 (ii) *G is a spanning subgraph of a $\prec$-augmentation $G_\prec = (V, E_Q \,\dot\cup\, E_P)$ such that $G_\prec$ has an augmented queue embedding with Hamiltonian path E_P and queue edges E_Q.*

(iii) *G is a spanning subgraph of a $\prec$-augmentation $G_\prec = (V, E_Q \,\dot\cup\, E_P)$, where $G_\prec$ has an embedding such that the embedding of its dual $G_\prec^* = (F, E_Q^* \dot\cup E_P^*)$ is an augmented queue embedding with Hamiltonian path E_Q^* and queue edges E_P^*.*

Theorem 4 shows that the FIFO principle of a queue is represented in the dual of queue graphs. The dual Hamiltonian path E_Q^* defines a linear ordering of the faces of the queue graph, which simultaneously reflects the linear ordering of insertions and of removals of data items. Accordingly, the $\prec$-augmentation of a stack graph $G = (V, E_S \,\dot\cup\, E_P)$ with stack edges E_S processed at the tail and Hamiltonian path E_P (drawn bold) yields a tree for the dual which is a (weak) dual of an outerplanar graph and reflects the LIFO principle (see Fig 14(a)). For the double-stack, the dual consists of two trees which are glued at their leaves and have a common root. The deque combines the stack with the queue and we obtain trees, for the stack edges, whose roots are connected by a path by the queue edges (see Fig. 14(b) for an example).

For another observation, let a graph $G = (V, E_Q \,\dot\cup\, E_P)$ be endowed with an augmented queue embedding. Then, $G^* = (F, E_Q^* \,\dot\cup\, E_P^*)$ is endowed with an

augmented queue embedding with Hamiltonian path Q^*, which is also Eulerian on E_Q^* as each edge of E_Q^* is traversed exactly once. Now, we remove an edge $e \in E_P$ from G. In the dual G^*, this is equivalent to contracting the dual edge e^* of e, i.e., the endpoints of e^* are identified. In the obtained dual graph, Q^* is not Hamiltonian anymore as one face is visited twice. However, Q^* is still Eulerian and it remains so after removing all edges of E_P from G. The obtained graph $G \setminus E_P$ is a queue graph and is endowed with a planar LC rotation system that corresponds to a queue layout.

Corollary 4.5 *If G is a queue graph with queue layout Σ such that G is embedded according to the corresponding LC rotation system $\Lambda = \Sigma$, then G^* contains a Eulerian path.*

5 Recognition Problems

The recognition problems have been raised and settled for stack, queue and deque graphs. Deciding whether a graph is a one-stack graph can be done in linear time as it is equivalent to testing outerplanarity [9, 35]. The other decision problems are $\mathcal{NP}$-complete, and were proved in [11, 36] for a double-stack, in [6] for a deque, and in [24] for a queue. For a double-stack and a deque, the $\mathcal{NP}$-hardness is due to the existence of a Hamiltonian cycle or path in a planar drawing.

A planar deque schedule Σ takes its viewpoint at the vertices and sees only the tail or the head of an edge but not the other endpoint. If the other endpoint is taken into account, then it defines an *edge classification* and partitions the set of edges into $E = E_{hh} \dot\cup E_{ht} \dot\cup E_{tt} \dot\cup E_{th}$, where, e.g., E_{hh} is the set of edges that leave and enter the deque at the head. Thereby, it classifies the edges into first and second stack edges E_{hh} and E_{tt}, and upward and downward queue edges E_{th} and E_{ht}, respectively. The situation is similar for a planar LC rotation system.

Recall from Section 3.1 that the fundamental polygon representation of an LC drawing can be seen as an extended two-level drawing with intra-level and inter-level edges [8, 26], so that the vertices on the upper level (at the tail) are clones of the vertices on the lower level (at the head), or vice versa. The queue edges are inter-level edges and the stack edges intra-level edges. The LC drawing is planar if and only if the two-level drawing is planar.

For the recognition problem of extended k-level graphs, we are given a partition of the vertices $V = V_1 \dot\cup \ldots \dot\cup V_k$ so that the vertices of V_i are placed on the i-th level, and an edge partition of the intra-level edges E_i between vertices of V_i into the sets of edges $E_i^a \dot\cup E_i^b$ above and below the i-th level. We search for k permutations $\pi_1, \ldots, \pi_k$, one for the vertices of each level, so that the extended k-level drawing obtained by ordering the vertices of V_i according to π_i is planar. This is the k-level planarity problem if there are no intra-level edges, and is solvable in linear time [29]. The extended k-level problem with intra-edges is solvable in linear time if, for every i, the subgraph induced by the set of edges below level i consists of a single connected component [26].

In contrast, the recognition problem for deque graphs asks for an edge partition and a single permutation for the vertices (on both sides of the fundamental polygon or two levels) such that the drawing is planar. This problem is $\mathcal{NP}$-complete [6], even if there are only inter-level edges [11, 36] or only intra-level edges [24].

In consequence, LC drawings should not be regarded as two-level drawings when the recognition problem is taken into account.

The recognition problem for deque graphs can be specialized such that either an edge partition or a linear ordering of the vertices, i. e., a Hamiltonian cycle (path), is given.

Lemma 4 *There is a linear time algorithm that computes the edge classification of a deque (double-stack) graph if the Hamiltonian path (cycle) is given.*

Proof: Since a deque graph G is planar, we run a dfs-based planarity test [12, 19, 28] so that the Hamiltonian path is the leftmost path in the initial dfs-tree. If there is a Hamilton cycle first remove one edge. Then the algorithm flips the remaining edges to the left or to the right of this path, in which case they are stack edges that enter the deque from one side, or the edges go from one side to the other, in which case they are queue edges, see also [5]. The planarity test algorithms run in linear time.   □

Note that the problem of minimizing the number of stacks for a k-stack layout of a graph is $\mathcal{NP}$-complete if the linear ordering of the vertices is fixed [24]. Thus the edge classification problem becomes hard if the objective is a minimum number of stacks, even if the linear ordering of the vertices is given.

The recognition problems reveals another distinction between deque and double-stack graphs. An instance of the *edge-partitioned deque graph recognition problem* consists of a graph $G = (V, E)$ and an edge partition $E = E_{hh} \dot\cup E_{ht} \dot\cup E_{tt} \dot\cup E_{th}$ and it asks for a planar deque layout such that E_{hh} is the set of edges that leave and enter the deque at the head, and similarly for E_{ht}, E_{th} and E_{tt}.

Theorem 5 *The edge-partitioned deque graph recognition problem is*

(i) $\mathcal{NP}$-*complete for deque graphs and for queue graphs,*

(ii) *solvable in linear time for double-stack graphs.*

Proof: (i) follows from the $\mathcal{NP}$-completeness of recognizing queue graphs [24], where $E = E_{ht}$, and the fact that queue graphs are deque graphs without stack edges. If there are only queue edges, then edges of E_{th} can be converted into edges of E_{ht}, since the deque must be empty after the removal of an upward queue edge and before the insertion of a downward queue edge, as shown in the proof of Lemma 1.

(ii) has been proved by Hong and Nagamochi [27] for the identical setting of 2-page book embedding with a given assignment of edges to pages.   □

6   Summary and Perspectives

We have introduced linear cylindric drawings as a new drawing technique to display the working principles of a stack, double-stack, queue and deque, respectively. There is a one-to-one correspondence between feasibility and planarity. Conversely, an invalid operation is directly visible as a crossing. We characterized the deque graphs as the graphs with a linear cylindric planar drawing and as the subgraphs of planar graphs with a Hamiltonian path. In comparison, the double-stack graphs are the graphs with a linear drawing in the plane and are the subgraphs of planar graphs with a Hamiltonian cycle. Hence, the gap between a deque and a double-stack is "small" and consists of a single edge from the first to the last vertex that either augments a Hamiltonian path to a Hamiltonian cycle or excludes winding edges around the cylinder. However, the gap is "large" for the recognition problem if an edge partition is given. The differences between deque and double-stack graphs are summarized in Table 1.

	double-stack	deque
linear layout on a surface	plane	cylinder
planar graph with Hamiltonian	cycle	path
edge-partitioned recognition problem	$\mathcal{O}(n)$	$\mathcal{NP}$-complete

Table 1: Differences between double-stack and deque graphs

Linear cylindric drawings are well-suited for queue layouts. Then all edges wind around the cylinder or connect vertices on opposite sides of the fundamental polygon by straight lines, and they do not cross. This makes them superior to common drawings with twists and to arched leveled-planar drawings [16, 18], as Fig. 1 illustrates. Queue graphs have the special property that the dual of their augmentation by a Hamiltonian path is of the same type and the dual of a queue graph has an Eulerian path.

Stack and queue layouts and stack and queue graphs were studied at many places. Yet, the relationship between stack and queue layouts is not fully understood. Heath and Rosenberg [24] showed that there is an outerplanar graph with seven vertices that is not a queue graph. It is an extended complete binary tree with edges between vertices on the same level. There was no example for the converse. Graph G from Fig. 15 is a subdivision of K_4 and thus not outerplanar, but it is a queue graph. Note that K_4 is neither a (1-)stack graph nor a (1-)queue graph. Hence, the classes of 1-stack graphs and 1-queue graphs are incomparable, and the class of 1-queue graphs is not minor-closed, whereas the class of 1-stack graphs is closed under taking minors. Heath and Rosenberg showed that every 1-stack graph has a 2-queue layout and every 1-queue graph has a 2-stack layout. The latter also follows from Corollary 4.3.

Recall that it is open whether stack-number is bounded by queue-number and conversely [18, 23], where the stack-number (queue-number) of a graph is the minimum k such that G has a k-stack layout (k-queue layout). The queue-number of planar graphs is bounded by $\mathcal{O}(\log^4 n)$ [14], whereas the stack-number

Figure 15: (a) A non-outerplanar graph and (b) its 1-queue layout.

is at most four [38].

Linear cylindric drawings can be used to represent input- and output restricted deques from [31]. A head (tail) of an output restricted deque does not allow removals at the head (tail). These restrictions simplify the drawings, since the set E_{th} (E_{ht}) of the edge partition is empty.

There are maximal planar graphs without a Hamiltonian cycle, such as the Goldner-Harary graph (Fig. 9(c)), and without a Hamiltonian path [25]. Hence, the deque (double-stack) is not powerful enough to capture all planar graphs. In [5] a splittable deque was introduced as an extension of a deque and it was shown that a graph is planar if and only if it has a layout in the splittable deque. The surplus is to split a deque into sub-deques, and the splits are controlled by a depth-first search tree.

The usefulness of our approach can also be exploited for the representation of mixed data structures, such as a stack and a queue. A stack-queue layout of a graph is given by a linear ordering of the vertices and a partition of the edges into stack edges and queue edges, which each have a planar layout. However, such graphs are no longer planar, since, e.g., $K_{3,3}$ admits such a representation, and our paradigm does not longer hold. Classes of graphs from mixed data structures are not well-understood. It is known that every graph has a 3-stack subdivision, a 2-queue subdivision, or a 1-stack 1-queue subdivision [18]. But 1-deque graphs are planar, and so are their subdivisions. This fact shows again that the stack and the queue modes of a deque restrict one another, and that a deque is not a double-ended queue, as the full name, given by D. E. Knuth [31], may suggest.

7   Acknowledgements

References

[1] G. A. Atneosen. *On the Embeddability of Compacta in N-books: Intrinsic and Extrinsic Properties*. Ph. D. Thesis, Department of Mathematics, Michigan State University, 1968.

[2] C. Auer. *Planar Graphs and their Duals on Cylinder Surfaces*. Dissertation, Universität Passau, Germany, January 2014. URL: `https://opus4.kobv.de/opus4-uni-passau/frontdoor/index/index/docId/226`.

[3] C. Auer, C. Bachmaier, F. J. Brandenburg, W. Brunner, and A. Gleißner. Plane drawings of queue and deque graphs. In U. Brandes and S. Cornelsen, editors, *Proc. 18th International Symposium on Graph Drawing, GD 2010*, volume 6502 of *LNCS*, pages 68–79. Springer, 2010. `doi:10.1007/978-3-642-18469-7_7`.

[4] C. Auer, C. Bachmaier, F. J. Brandenburg, and A. Gleißner. Classification of planar upward embedding. In M. van Kreveld and B. Speckmann, editors, *Proc. 19th International Symposium on Graph Drawing, GD 2011*, volume 7034 of *LNCS*, pages 415–426. Springer, 2011. `doi:10.1007/978-3-642-25878-7_39`.

[5] C. Auer, F. J. Brandenburg, A. Gleißner, and K. Hanauer. Characterizing planarity by the splittable deque. In S. Wismath and A. Wolff, editors, *Proc. 21st International Symposium on Graph Drawing, GD 2013*, volume 8242 of *LNCS*, pages 25–36. Springer, 2013. `doi:10.1007/978-3-319-03841-4_3`.

[6] C. Auer and A. Gleißner. Characterizations of deque and queue graphs. In P. Kolman and J. Kratochvíl, editors, *Proc. 37th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2011*, volume 6986 of *LNCS*, pages 35–46. Springer, 2011. `doi:10.1007/978-3-642-25870-1_5`.

[7] C. Bachmaier, F. J. Brandenburg, and M. Forster. Track planarity testing and embedding. In P. Van Emde Boas, P. J., and S. J. Bielikova M. and, editors, *Proc. 31st SOFSEM*, volume 2 of *Theory and Practice in Computer Science*, pages 3–17. MatFyzPres, 2004.

[8] C. Bachmaier, H. Buchner, M. Forster, and S. Hong. Crossing minimization in extended level drawings of graphs. *Discrete Applied Mathematics*, 158(3):159–179, 2010. `doi:10.1016/j.dam.2009.09.002`.

[9] F. Bernhart and P. C. Kainen. The book thickness of a graph. *J. Combin. Theory, Ser. B*, 27(3):320–331, 1979. `doi:10.1016/0095-8956(79)90021-2`.

[10] F. J. Brandenburg. Upward planar drawings on the standing and the rolling cylinders. *Computational Geometry*, 47(1):25–41, 2014. `doi:10.1016/j.comgeo.2013.08.003`.

[11] V. Chvátal. *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization.* John Wiley and Sons, New York, NY, USA, 1985.

[12] H. de Fraysseix and P. O. de Mendez. Trémaux trees and planarity. *Eur. J. Comb.*, 33(3):279–293, 2012. `doi:10.1016/j.ejc.2011.09.012`.

[13] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice Hall, 1999.

[14] G. Di Battista, F. Frati, and J. Pach. On the queue number of planar graphs. *SIAM J. Comput.*, 42(6):2243–2285, 2013. `doi:10.1137/130908051`.

[15] G. Di Battista and E. Nardelli. Hierarchies and planarity theory. *IEEE Trans. Syst., Man, Cybern.*, 18(6):1035–1046, 1988. `doi:10.1109/21.23105`.

[16] V. Dujmović and D. R. Wood. On linear layouts of graphs. *Discrete Math. Theor. Comput. Sci.*, 6(2):339–358, 2004. URL: `http://dmtcs.episciences.org/317`.

[17] V. Dujmović and D. R. Wood. Three-dimensional grid drawings with subquadratic volume. In G. Liotta, editor, *Proc. 12th Symposium on Graph Drawing, GD 2004*, volume 2912 of *LNCS*, pages 190–201. Springer, 2004. `doi:10.1007/978-3-540-24595-7_18`.

[18] V. Dujmović and D. R. Wood. Stacks, queues and tracks: Layouts of graph subdivisions. *Discrete Math. Theor. Comput. Sci.*, 7(1):155–202, 2005. URL: `http://dmtcs.episciences.org/346`.

[19] S. Even. *Graph Algorithms.* Cambridge University Press, New York, NY, USA, 2nd edition, 2012.

[20] A. Goldner and F. Harary. Note on a smallest nonhamiltonian maximal planar graph. *Bull. Malaysian Math. Soc.*, 6(1):41–42, 1975.

[21] S. M. Hashemi. Digraph embedding. *Discrete Math.*, 233(1–3):321–328, 2001. `doi:10.1016/S0012-365X(00)00249-1`.

[22] P. Healy, A. Kuusik, and S. Leipert. A characterization of level planar graphs. *Discrete Math.*, 280(1–3):51–63, 2004. `doi:10.1016/j.disc.2003.02.001`.

[23] L. S. Heath, F. T. Leighton, and A. L. Rosenberg. Comparing queues and stacks as mechanisms for laying out graphs. *SIAM J. Discret. Math.*, 5(3):398–412, 1992. `doi:10.1137/0405031`.

[24] L. S. Heath and A. L. Rosenberg. Laying out graphs using queues. *SIAM J. Comput.*, 21(5):927–958, 1992. `doi:10.1137/0221055`.

[25] G. Helden. *Hamiltonicity of Maximal Planar Graphs and Planar Triangulations.* Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, Germany, 2007. URL: `http://darwin.bth.rwth-aachen.de/opus3/volltexte/2007/1949/`.

[26] S.-H. Hong and H. Nagamochi. Testing planarity of level graphs with intra-level edges. Technical report, Department of Applied Mathematics and Physics, University of Kyoto, Japan, 2009. URL: `http://www.amp.i.kyoto-u.ac.jp/tecrep/abst/2009/2009-005.html`.

[27] S.-H. Hong and H. Nagamochi. Simpler algorithms for testing two-page book embedding of partitioned graphs. In Z. Cai, A. Zelikovsky, and A. G. Bourgeois, editors, *Computing and Combinatorics, COCOON 2014*, volume 8591 of *LNCS*, pages 477–488. Springer, 2014. `doi:10.1007/978-3-319-08783-2_41`.

[28] J. E. Hopcroft and R. E. Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974. `doi:10.1145/321850.321852`.

[29] M. Jünger and S. Leipert. Level planar embedding in linear time. *J. Graph Alg. App.*, 6(1):67–113, 2002. `doi:10.7155/jgaa.00045`.

[30] P. C. Kainen. Some recent results in topological graph theory. In R. A. Bari and F. Harary, editors, *Graphs and Combinatorics*, volume 406 of *LNM*, pages 76–108. Springer, 1974. `doi:10.1007/BFb0066436`.

[31] D. E. Knuth. *The Art of Computer Programming, Volume I: Fundamental Algorithms.* Addison-Wesley, Reading, Massachusetts, MA, USA, 1968.

[32] L. T. Ollmann. On the book thicknesses of various graphs. In *Proc. 4th Southeastern Conference on Combinatorics, Graph Theory and Computing*, volume 8 of *Congressus Numerantium*, page 459, 1973.

[33] H. Petersen. Stacks versus deques. In J. Wang, editor, *Proc. 7th Annual International Conference on Computing and Combinatorics, COCOON 2001*, volume 2108 of *LNCS*, pages 218–227. Springer, 2001. `doi:10.1007/3-540-44679-6_24`.

[34] W. T. Tutte. How to draw a graph. *Proc. Lon. Mat. Soc.*, 13:743–767, 1963.

[35] M. Wiegers. Recognizing outerplanar graphs in linear time. In G. Tinhofer and G. Schmidt, editors, *Proc. 12th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 1986*, volume 246 of *LNCS*, pages 165–176. Springer, 1987. `doi:10.1007/3-540-17218-1_57`.

[36] A. Wigderson. The complexity of the Hamiltonian circuit problem for maximal planar graphs. Technical Report 298, Department of EECS, Princeton University, February 1982.

[37] D. R. Wood. Queue layouts, tree-width, and three-dimensional graph drawing. In M. Agrawal and A. Seth, editors, *Proc. Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2002*, volume 2556 of *LNCS*, pages 348–359. Springer, 2002. `doi:10.1007/3-540-36206-1_31`.

[38] M. Yannakakis. Embedding planar graphs in four pages. *J. Comput. Syst. Sci.*, 38(1):36–67, 1989. `doi:10.1016/0022-0000(89)90032-9`.