PLANAR GRAPH DRAWING
2.1 Introduction

Visualization of an arbitrary large graph $G = (V, E)$, where $V$ is the set of participating vertices and $E$ is the set of the edges, is a tedious process and the shape of the graph becomes unpredictable if no systematic procedures are followed. De Fraysseix, Pach and Pollack [4] relied upon triangulation of the graph as a first step to visualize a graph. Chrobak and Payne [5] improved this and presented an algorithm as a generalization to the triangulation. These methods tend be problematic in realizing a decent looking large graph on the plane. We have improved the algorithm and it embeds a bi-connected graph of $n$ vertices on a grid of size $(2n-4) \times (n-2)$ in linear time on a plane. This new form of the graph is topologically equivalent to the original graph and facilitates further study.

2.2 Chrobak & Payne Algorithm for Drawing of Planar Graphs

The algorithm has two steps: the first step calculates the canonical ordering of the vertices (which is the order in which the vertices will be processed), and the second step then constructs the drawing incrementally, adding vertices to the current drawing one by one according to the canonical ordering thus induced.

![Fig. 2.1 Sample output of the Chrobak & Payne Algorithm on a Non-Triangulated Graph](image-url)
2.2.1 Canonical Ordering

Let $G$ be a planar graph drawn on the plane. Let $u, v, w$ be vertices on the boundary of its exterior face. The canonical ordering is a labeling of the vertices of $G$ in a sequence $v_1, v_2, ..., v_n$ such that $v_1 = u$, $v_2 = v$, $v_n = w$ and for every $3 \leq k \leq n$ the following hold:

(a) the subgraph $G_{k-1}$ of $G$ induced by $v_1, v_2, ..., v_{k-1}$ is bi-connected, and the boundary of the exterior face is a cycle $C_{k-1}$ containing the edge $(u, v)$.

(b) The vertex $v_k$ is on the exterior face of $G_{k-1}$, and has at least two neighbours in $G_{k-1}$. Moreover, all of its neighbours in $G_{k-1}$ are consecutive on the path $G_{k-1} - (u, v)$.

![Fig. 2.2 A Canonically Ordered Vertices Graph](image)

Fraysseix, Pach and Pollack [4] proved that such an ordering exists for all planar graphs for planar embedding. A planar embedding is a data structure that describes the circular ordering of neighbours of each vertex in some planar
drawing. The ordering algorithm works by processing each of the vertices and its neighbours. Vertices are labeled and the labels are updated when visiting the neighbours. The labels assume numerical values as under:

-1, initially assigned, meaning not yet visited
0, meaning visited once
t >0, meaning visited more than once and those of its neighbours already visited form t intervals in the circular order around the vertex given by the planar embedding.

We start by choosing two vertices, calling them $v_1$ and $v_2$. Processing a vertex $v_k$ is always carried out by visiting each of its neighbours and updating the labels of those neighbours that are not yet processed. Let $v$ be neighbour of $v_k$. Then the possibilities are as under:

case (i): $v$ is labeled $-1$, in which case we re-label it as $0$.

case (ii): $v$ is labeled as $0$. This means that $v$ has one neighbour, say $u$, which has already been processed. Check if $v_k$ is adjacent to $u$ in the circular ordering of neighbours around $v$ (given by the planar embedding). If so, label $v$ with $1$, else label it with $2$.

case (iii): $v$ is labeled $t > 0$. Check the two vertices adjacent to $v_k$ in the circular ordering around $v$. If both have already been processed, label $v$ with $t-1$ (i.e., two intervals have now been merged). If one has been processed and the other not then $v$'s label remains $t$. If have not been processed, label $v$ with $t+1$. 
After processing $v_k$ (for $k \geq 2$), a vertex with label 1 is chosen to be $v_{k+1}$ in the canonical ordering (any such vertex serves the purpose), and is thereby processed. This continues until no such $v_{k+1}$ is found.

### 2.2.2 The Placement Step

The second phase of the algorithm places the vertices on the grid points of a chosen mesh. Let $Q = (x_1, y_1)$, $R = (x_2, y_2)$ be two grid points and $\mu(Q, R)$ be the intersection point of the lines with slope +1 from $Q$ and slope -1 from $R$. That is:

$$\mu(Q,R) = \frac{1}{2}(x_1 - y_1 + x_2 + y_2, -x_1 + y_1 + x_2 + y_2).$$

Define the norm between $Q$ and $R$ as $MD(Q,R) = |x_2 - x_1| + |y_2 - y_1|$. If the norm is even, then $x_1 - y_1 + x_2 + y_2$ and $-x_1 + y_1 + x_2 + y_2$ are also even. This means that if $Q$ and $R$ are grid points with even norm, then $\mu(Q,R)$ must be a grid point too. Let $P(v) = (x(v), y(v))$ denote the current position of vertex $v$ on the grid. With each such $v$ associate a set $L(w)$ as follows:

- $P(v_1) \leftarrow (0,0)$; $L(v_1) \leftarrow \{v_1\}$;
- $P(v_2) \leftarrow (2,0)$; $L(v_2) \leftarrow \{v_2\}$;
- $P(v_3) \leftarrow (1,1)$; $L(v_3) \leftarrow \{v_3\}$;

$v_1$, $v_2$, $v_3$ are placed on a triangle. The subgraph $G_k$ is formed by adding vertex $v_k$ to $v_1$, $v_2$, ..., $v_{k-1}$ at each step. At any $k^{th}$ step of the algorithm, the contour $C_k$ of $G_k$, will be of triangle like shape, and the following properties hold:

1. $P(v_1) \leftarrow (0,0)$ and $P(v_2) \leftarrow (2k-4,0)$
2. $C_k = w_1, w_2, ..., w_m$ for some $m$, where $w_1 = v_1$, $w_m = v_2$ and $x(w_1) < x(w_2) < ... < x(w_m)$.
3. The slope of each segment $(P(w_i), P(w_{i+1}))$ for $1 \leq i < m$, is either $\pm 1$. 


Assume that this is true upto k-1 steps. We will now add $v_k$ to the drawing. By canonical ordering, we can assume that $v_k$ is such that its neighbours on $C_{k+1}$ are consecutive, and we can denote them by $w_p, ..., w_q$. The inclusion can be accomplished with the following scheme:

for each $v \in \bigcup_{i=p}^{q-1} L(w_i)$ do $x(v) \leftarrow x(v) + 2$; (i.e., move such $v$ to right by 2)

for each $v \in \bigcup_{i=p+1}^{q-1} L(w_i)$ do $x(v) \leftarrow x(v) + 1$; (i.e., move such $v$ to right by 1)

$P(v_k) \leftarrow \mu(P(w_p), P(w_q))$

$L(v_k) \leftarrow \{v_k\} \bigcup \bigcup_{i=p+1}^{q-1} L(w_i)$

Now, by (p3) we know that if $w_i$ and $w_j$ are any two vertices on the contour, and $I = P(w_i)$ and $J = P(w_j)$ are their current positions on the grid, then $MD(I, J)$ is even. As a result of this, $\mu(P(w_p), P(w_q))$ is always a grid point. We can ensure that all $v_k$'s neighbours will be visible from $P(v_k)$ by moving some of the points $P(w_i)$ to the right. With each vertex $v$ that moves we also move the set $L(v)$, consisting of the vertices that reside below it. This is needed to keep the part that has already been drawn without crossings intact in shape. The sets $L(v)$ can be binary trees rooted at $v$. At step $k$, the offspring of $v_k$ are the vertices $w_{p+1}, ..., w_{q-1}$, which are the roots of the trees $L(w_{p+1}), ..., L(w_{q-1})$. The left-child holds the first offspring of a vertex, and the right-child holds the first sibling to the right of the vertex. To achieve constant time for updating this structure at step $k$, the contour chain is kept in the right-child array and we continue with the following checks:

if $w_{q-1} \neq w_p$ then $\text{leftchild}(v_k) \leftarrow w_{p+1}$ else $\text{leftchild}(v_k) \leftarrow \text{null}$

if $w_{q-1} \neq w_p$ then $\text{rightchild}(w_{q-1}) \leftarrow \text{null}$
All other right-child connections are inherited automatically from the contour. The contour chain is updated with rightchild \((w_p) \leftarrow v_k\)
rightchild \((v_k) \leftarrow w_q\). The calculation of the x coordinate of \(v_k\) is carried out relative to that of \(w_p\), and at the end of the algorithm these relative coordinates are translated into real ones by a single traversal of the binary tree. Since the vertices of the graph are processed according to their canonical ordering, a planar drawing is guaranteed.

2.3 Suggested Refinement

First we note that the two requirements of the canonical ordering cannot be fulfilled when the graph is not triangulated, and the graph \(G_k\) needs to be bi-connected. However, if we take a cycle on \(n\) vertices as input, any possible \(G_{n-1}\) will be a path that is not bi-connected. Secondly, it requires \(v_{k+1}\) to have consecutive neighbours on the path \(C_k = (v_1, v_2)\).

![Fig. 2.3 Example of a non-triangulated graph](image_url)

However, consider Fig 2.3 with exterior face is \(v_1, v_2, x\). Here the canonical ordering must have \(v_4 = x\) implying \(C_3 = v_1, y, v_2\). But the neighbours of \(v_4\) do not form a consecutive interval on \(C_2\) as \(y\) is not a neighbour of \(x\). So we need a
broader meaning for canonical ordering, to enable us to draw a graph vertex by vertex in the placement step and it may be called bi-connected canonical ordering.

2.3.1 Right Hand Walk On Regions

Given an orientation for the edge \((u,v)\) and \((v,u)\), both of which are indicated by the line segment \(uv\), we speak of the right face and left face of \((u,v)\) on a walk with the region on right or left. Note that these might be the same as in the case where \(v\) has no incident edges other than \((u,v)\). The boundary of each face in the drawing consists of a single connected polygonal line. We will produce a boundary list for each face as under:

2.3.2 Procedure RightHandWalk:

Mark all edges of \(G\) as unvisited;

While there are unvisited edges do the following

Choose any unvisited edge \((u,v)\) to initialize a new list \(b\) with \(v_0 = u\) and \(v_1 = v\)

Set \(i \leftarrow 1\)

Repeat

Take as \(v_{i+1}\) the vertex immediately following \(v_{i-1}\) in the counter-clockwise circular ordering of neighbours around \(v_i\)

Add \(v_{i+1}\) to the list \(b\)

Mark the edge \((v_i, v_{i+1})\) as visited

Set \(i \leftarrow i + 1\)

Until \((v_i, v_{i+1}) = (v_0, v_1)\)

Close the list \(b\)

End-while

The whole process can be viewed as a person walking along the edges of the graph choosing the rightmost option at every vertex. The resulting list \(b(f)\)
represents the boundary of a face $f$ in a clockwise direction. Clearly the right face of a directed edge $(u,v)$ is also the left face of the dual edge $(v,u)$.

Thus if $f$ is a face and $b(f) = v_0, v_1, ..., v_m$ is the list produced by the right hand walk, the reversed list is another representation of the boundary of $f$, traversing it in a counter-clockwise fashion, and $f$ is the left face of each of the edges $(v_1, v_{i-1})$. We refer to the reversed lists as counter-clockwise boundary lists, or just boundary lists for short. Similarly we define a LeftHandWalk algorithm.

2.3.3 Cut-Vertex

If in a Graph $G$, a vertex $v$, after removal of its edges splits the Graph $G$ into unconnected subgraphs then $v$ is called a cut-vertex of $G$. Each directed edge appears in exactly one boundary list. An undirected edge might appear in two different boundary lists, once in each direction, or it might appear in the same boundary list in both directions. As far as vertices go, unless $v$ is a cut-vertex of $G$, it appears at most once in each boundary list.

If $v$ is a cut-vertex, each one of the boundary lists corresponding to the components of $G$ that include $v$ will contain $v$ more than once. To construct the boundary lists, we do not need the planar drawing itself – all we need is a planar embedding as we only use the circular ordering of neighbours around each vertex.

2.3.4 Bi-Connected Canonical Ordering

Let $G$ be a bi-connected planar graph drawn in the plane. Let $G_k$ be a connected subgraph of $G$ and let $C_k = w_1, w_2, ..., w_m$ be the counter-clockwise boundary list of the exterior face of $G_k$. Let $v$ be a vertex in $G - G_k$ that lies in
the exterior face of $G_k$. Since $G$ is planar, that neighbour must lie on $C_k$ and we can thus assume it is $w_i$ for some $i, 1 \leq i \leq m$. We say that $v$ has a right support if $v$ immediately follows $w_{i+1}$ in the counter-clockwise circular ordering around $w_i$; it has a left support if $v$ immediately precedes $w_{i-1}$ in the counter-clockwise circular ordering around $w_i$.

Further $v$ has a legal support on $C_k$ if $i = 1$ and $v$ has a right support, or $i = m$ and $v$ has a left support, or $1 < i < m$ and $v$ has a left support or a right support. We observe that since $C_k$ is cyclic in nature, the starting point of the list, $w_1$, can be fixed arbitrarily along $C_k$.

A bi-connected canonical ordering is a labeling of the vertices of $G$ in a sequence $v_1, \ldots, v_n$, such that $v_1 = u$ and $v_2 = v$, and for every $2 \leq k \leq n$ the following hold:

(a) Let $G_k$ be induced by $v_1, \ldots, v_k$. Then $G_k$ is connected, and the edge $(v_2, v_1)$ is on $C_k$, the contour of $G_k$. Fix $w_1$ to be $v_1$, so that we write $C_k$ as $v_1, w_1, \ldots, w_m = v_2$.

(b) All vertices in $G - G_k$ lie within the exterior face of $G_k$.

(c) For $k > 2$, the vertex $v_k$ has one or more neighbours in $G_{k-1}$. If $v_k$ has exactly one neighbour in $G_{k-1}$, then it has a legal support on $C_{k}$.

We now device a scheme to find a bi-connected canonical ordering.

Let $C_{k-1}$ be the contour of $G_{k-1}$ induced by $v_1 = w_1, w_2, \ldots, w_m = v_2$. Let $v$ be a vertex outside $G_{k-1}$ with a neighbour in $G_{k-1}$. Then $v$ is in the exterior face
of $G_{k-1}$, and therefore lies on the exterior boundary of the larger graph $G_{k-1} \cup v$. Since $G_{k-1}$ is connected, $v$ is not a cut-vertex of $G_{k-1} \cup v$, and it therefore appears exactly once along the boundary list of the exterior face of $G_{k-1} \cup v$. By the planarity of $G$, the neighbours of $v$ in $G_{k-1}$ must all reside on $C_{k-1}$, and thus $u$ is really $w_{i_1}$ for some appropriate $1 \leq i_1 \leq m$. If $v$ has $p$ neighbours on $C_{k-1}$, we can list them similarly in their counter-clockwise circular ordering around $v_1$ as $w_{i_1}, w_{i_2}, ..., w_{i_p}$. A vertex $x$ may appear more than once in the list $C_{k-1}$. This could happen if it is a cut vertex of the graph $G_{k-1}$, and $C_{k-1}$ goes around a component attached to $x$.

Thus, if $x$ is one of the neighbours of $v$, we should be more precise in defining the index $i_j$ that satisfies $x = w_{i_j}$. Obviously there is one index $q$ that satisfies $x = w_q$, and such that on the clockwise circular ordering around $x$ the order is $w_{q-1}, v, w_{q+1}$. This $q$ serves as $i_j$. Suppose $w_{i_1}, w_{i_2}, ..., w_{i_p}$ be the neighbours of $v$ on the contour $C_{k-1} = w_1, w_2, ..., w_m$ as defined above. Then $i_1 < i_2 < ... < i_p$.

An example is given in Fig.2.4. The meaning is that the circular ordering around $v$ coincides with the order along the boundary list $C_{k-1}$. Bearing in mind that boundary lists are circular in nature, the particular starting point $w_1$, chosen for $C_{k-1}$, the list of neighbours of $v$ on $C_{k-1}$ does not wrap around the circular list $C_{k-1}$.
2.4 Requirements for Canonical Ordering of Vertices

We employ three arrays $A$, indexed by the faces of the graph; $N$ and $F$, indexed by the vertices. At the $k$th stage, $A(f)$ contains the number of edges from $b(f)$ that are in $G_{k-1}$; $N(v)$ contains the number of neighbours of vertex $v$ in $G_{k-1}$, and $F(v)$ represents the number of \textit{ready} faces that have $v$ as their only vertex outside $G_{k-1}$. A face $f$ that is not the exterior face of $G$ is \textit{ready} if $A(f) = |b(f)| - 2$, i.e., $b(f)$ has only two edges not in $G_{k-1}$. Let $v \notin G_{k-1}$. Denote $N(v)$ by $p$, and let $w_{i_1}, w_{i_2}, \ldots, w_{i_p}$ be the neighbours of $v$ on $C_{k-1}$. Also, let $f_j$ be the left face of the edge $(w_{i_j}, v)$, for $1 \leq j \leq p$ and $L_v$ be the circular ordering of all neighbours of $v$. 

Fig 2.4. Illustration for Ordered Traversal – Canonically Ordered Vertices
2.4.1 Properties of Number of Neighbours $N(v)$

(i) $N(v) > F(v)$.

(ii) $N(v) = F(v) + 1$ if and only if all the faces $f_j$, for $2 \leq j \leq p$, are ready.

(iii) If $N(v) = F(v) + 1$, then the neighbours of $v$ in $G_{k-1}$ form a single interval in the list $L_v$.

Proof:

(i). $F(v)$ contains the ready faces that have $v$ as their sole vertex outside $G_{k-1}$. The boundary list of such a face has an edge of the form $(w_{i_j}, v)$, with $w_{i_j}$ in $G_{k-1}$. Thus each of the ready faces is the left face of one of the edges $(w_{i_j}, v)$, such that the set of ready faces found for $F(v)$ is a subset of the $p$ faces $f_1, f_2, \ldots, f_p$. Hence, $N(v) = p \geq F(v)$. To prove that $N(v) > F(v)$, we will show that $f_1$ cannot be a ready face. Note that $b(f_1)$ is the boundary list of $f_1$ on $G_k$, so that $b(f_1)$ contains the edge $(w_{i_1}, v)$. If $b(f_1)$ has only $v$ as a vertex not in $G_{k-1}$, it is entirely contained in $G_{k-1} \cup v$. By our choice of $w_{i_1}$, the left face of the edge $(w_{i_1}, v)$ in the subgraph $G_{k-1} \cup v$ is the exterior face of this subgraph. Now $G_{k-1} \cup v$ contains the edge $(v_2, v_1)$, which is on the boundary of the exterior face of the entire graph. Thus $(v_2, v_1)$ is on the boundary of the exterior face of the subgraph $G_{k-1} \cup v$ too, and it therefore belongs to $b(f_1)$. However, in the entire graph $G$, the face whose boundary list contains the edge $(v_2, v_1)$ is the exterior face, which means that $f_1$ must be the exterior face of $G$. The
exterior face of the entire graph $G$ was excluded from the definition of a ready face. Hence $f_1$ cannot be a ready face.

(ii). Assume $N(v) = F(v)+1$. The above counting shows that each of the $p-1$ faces $f_2, \ldots, f_p$ must be ready. Conversely, since the set of faces accounted for in $F(v)$ consists of those faces from among $f_2, \ldots, f_p$ that are ready, then if they are all ready we must have $F(v) = p-1$, which is $N(v) = F(v)+1$.

(iii). Let $N(v) = F(v)+1$, and assume that $L_v$ contains a fragment of the form $w_{i_{j-1}}, \ldots, u, w_{i_j}$ for some $2 \leq j \leq p$, meaning that there are vertices that separate a pair of adjacent neighbours of $v$ in $G_{k-1}$. We recall that $w_{i_1}, w_{i_2}, \ldots, w_{i_p}$ is the list of neighbours of $v$ in $G_{k-1}$, ordered counterclockwise around $v$. Thus $u \not\in G_{k-1}$ (otherwise it would be one of $w_{i_{j-1}}$ or $w_{i_j}$). Now, since $u$ follows $w_{i_j}$ in the clockwise circular ordering around $v$, the boundary list $b(f_j)$ must contain the edges $(w_{i_j}, v)$ and $(v, u)$. This implies that $f_j$ has two vertices outside $G_{k-1}$ which are $v$ and $u$, and therefore it cannot be a ready face. This contradicts the assumption that $N(v) = F(v)+1$, thus completing the proof.

2.5 Refined Algorithm for Canonical Ordering of Vertices

The arrays are updated always as under:

(1) **Update $v_k$'s neighbours.** For each neighbour $v$ of $v_k$ that is outside $G_k$, increment $N(v)$ by 1.

(2) **Update faces.** There are two faces to update. The left face of the edge $(w_{i_1}, v_k)$, which is $f_1$, and the right face of $(w_{i_p}, v_k)$, which we shall call $f_{p+1}$. For these increment $A(f_1)$ and $A(f_{p+1})$ by 1.
(We recall that \( w_{i_1}, w_{i_2}, \ldots, w_{i_p} \) is the ordered list of neighbours of \( v_k \) on \( C_{k-1} \). Also, it might be the case that \( f_1 = f_{p+1} \).)

(3) Update ready faces. If a face \( f \) becomes ready as a result of (2), find the only vertex \( v \) along \( b(f) \) that is outside \( G_k \), and increment \( F(v) \) by 1.

2.5.1 Algorithm for building the new bi-connected canonical ordering

Initialize all three arrays, \( A \), \( N \) and \( F \) to 0.

Take as \( (v_1, v_2) \) any edge on the boundary of the exterior face of \( G \).

Initialize a list of vertices with \( v_1 \) and \( v_2 \), and update their neighbours as in (1) above.

Set \( A(f) \) to 1 for \( f \), the left face of \( (v_1, v_2) \).

If \( f \) is a triangle with vertices \( v_1, v_2, v_3 \), set \( F(v_3) \) to 1, since \( f \) is ready.

For \( k = 3 \) to \( n \) do the following

If there is a vertex \( v \) not in the list, with \( N(v) \geq 2 \) & \( N(v) = F(v) + 1 \) then

add it to the list as \( v_k \)

else

find a vertex \( v \) not on the list, with legal support and \( N(v) = 1 \), and add it to the list as \( v_k \)

Update the data structures as in (1),(2),(3) above for \( v_k \).

end-for.
2.5.2 Further Observations

- The vertex $v_k$ is drawn only after at least one of its neighbours has already been drawn in $G_{k-1}$. This prevents the situation where concave polygons might appear in the drawn diagram.

- The drawing is generally bound by $(2n-4) \times (n-2)$ size grid. The algorithm starts the placement step with the edge $(v_1, v_2)$ drawn with length 2, and it increases this length by 2 at each step, ending with length $(2n-4)$. The entire drawing can be enclosed in a triangle whose base is the edge $(v_1, v_2)$, and whose sides emanate from $v_1$ and $v_2$ with slopes +1 and -1 respectively. Hence the drawing's maximum height is $(n-2)$.

- Our canonical ordering requires the graph to be bi-connected. There are examples of planar non-biconnected graphs for which no bi-connected
canonical orderings exist. However, every graph can be made bi-connected by adding dummy edges using the following scheme:

- Given any two disconnected components, add a dummy edge to connect arbitrarily chosen vertices, in each of them.
- Given two components with a common cut-vertex $v$, add a dummy edge that connects arbitrary neighbours of $v$, one from each component.

Deciding connectivity and bi-connectivity, and identifying bi-connected components can all be done in linear time. Once we draw the graph after the completion of the placement step, we remove the dummy edges.

### 2.6 Analysis of the Algorithm

For each $2 \leq k \leq n$, let $v_1, v_2, \ldots, v_k$ be the sequence of vertices generated by the algorithm up to the stage $k$. Then the conditions (a),(b),(c) in the definition of bi-connected canonical ordering are satisfied, and also there exists a vertex $v$ outside $G_k$, such that either $N(v) \geq 2$ and $N(v)=F(v)+1$, or $N(v) = 1$ and $v$ has a legal support. Thus the algorithm is capable of drawing the graph completely.

We apply induction on $k$. For $k=2$, $G_2$ consists of the single edge $(v_1, v_2)$. All the conditions are trivially clear. Now assume that we have built the sequence $v_1, v_2, \ldots, v_{k-1}$ such that conditions (a),(b),(c) hold. The graph $G$ is connected, so there are vertices outside $G_{k-1}$ with $N(v)>0$. First assume that all these vertices have $N(v)=1$. Let $v$ be such a vertex, and let $w_i$ be its sole neighbour on $C_{k-1}$. If $v$ itself does not have a legal support, then the vertex $t_1$ that immediately follows $w_{i+1}$ on the counter-clockwise circular ordering of neighbours around $w_i$ satisfies $N(t_1)=1$ and has a right support, and the vertex $t_2$ that immediately precedes $w_{i-1}$,
satisfies $N(t_2)=1$ and has a left support. In this case, one of $t_1$ or $t_2$ must have a legal support, and it can therefore be chosen to be $v_k$.

Fig. 2.6 Illustration for Building of Legal Support

Let $v$ be a vertex outside $G_{k-1}$, which has $p$ neighbours in $G_{k-1}$. As before, we let $C_{k-1} = w_1, w_2, \ldots, w_m$, where $w_1 = v_1$ and $w_m = v_2$. We denote the $p$ neighbours of $v$ by $w_{i_1}, w_{i_2}, \ldots, w_{i_p}$ as they constitute a sub-series of $C_{k-1}$. Let us call the fragment of $C_{k-1}$ between $w_{i_1}$ and $w_{i_p}$ as the span of $v$ on $C_{k-1}$. The length of the span is $i_p - i_1$ (which is the number of vertices therein).

Now let there be some vertices in the graph with $N(v) \geq 2$, and assume, for contradiction, that none of these vertices satisfies $N(v) = F(v) + 1$. Let $v$ be a vertex with $N(v) \geq 2$ whose span on $C_{k-1}$ is the shortest among the vertices with $N(v) \geq 2$. Let $w_{i_1}, w_{i_2}, \ldots, w_{i_p}$ be a span of $v$ on $C_{k-1}$. By our assumption, $v$ satisfies $N(v) = F(v) + 1$, implying that one or more of these faces $f_2, \ldots, f_p$ is not ready. Let $f_{j+1}$, for some $1 \leq j \leq p-1$, be one if these, and let $H$ be the subgraph of $G$ induced by the closed polygonal line. Let $e$ be the left face of the edge $(w_{i_{j+1}}, v)$ in the subgraph $H$. If $e$ was a face in the original graph $G$ too, it must have been ready,
since it has only two edges outside $G_{k-1}$. However, by our assumptions, this is impossible. Consequently the region $e$ in $G$ is not connected, and hence there is a path $P$ in $G$ that divides $e$ in two.

We claim that $P$ cannot be a single edge. First, if $P$ were an edge connecting $v$ to some vertex $w$ on $C_{k-1}$, then $w$ must be between $w_{ij}$ and $w_{ij+1}$, contradicting the order in the span of $v$. Second, if $P$ were an edge connecting two $w_i$'s, then $P$ would belong to the subgraph $G_{k-1}$, and as such, it could not be inside the exterior face of $G_{k-1}$. This is impossible due to the induction hypothesis that $v$ is in the exterior face of $G_{k-1}$, so that the region $e$ (which includes $P$) is in that exterior face too. Thus, $P$ must have at least one internal vertex, call it $x$. Since we assume that $G$ is bi-connected, there is a path $Q$ in $G$ that connects $x$ and $w_{ij}$ without passing through $v$ (otherwise $v$ is a cut-vertex of $G$). Let $w_1$ be the first point on $Q$ (when coming from $x$) that is in $G_{k-1}$, and let $y$ be the point preceding $w_1$ on $Q$. Since $w_1$ is on the boundary of $e$ and $x$ is in $e$, $y$ must also be in $e$.

![Fig.2.7 Illustration for Building of Legal Support](image)
Now, if \( y \) satisfies \( N(y) > 1 \), we have found in \( y \) a contradiction to the minimality assumption on \( v \). Here is why: First, we claim that \( y \)'s span is part of \( v \)'s. To see this, note that every edge emanating from \( y \) must reside entirely in \( e \) (by planarity), and it therefore can only lead to points in \( e \) or on its boundary. The intersection of the boundary of \( e \) with \( C_{k-1} \) is the sequence \( S = w_{i_1}, \ldots, w_{i_{k-1}} \), and hence \( y \)'s span can only be a sub-sequence of \( S \), while \( S \) itself is part of \( v \)'s span. This implies that the length of \( y \)'s span cannot exceed the length of \( v \)'s; if it is strictly smaller, we have a contradiction to the choice of \( v \). If \( v \) and \( y \) happen to have exactly the same span, the circular ordering around \( w_i \) (taken counterclockwise) must be \( w_{i+1}, \ldots, y, \ldots, v \), which contradicts the second part of the minimality assumption on \( v \).

If \( N(y) = 1 \), so that \( y \) has only one neighbour on \( C_{k-1} \), we will find a contradiction. Again we have two cases to consider. If \( w_1 \neq w_{i_j} \), let \( z \) be the neighbour of \( w_1 \) immediately following \( w_{i-1} \) along the clockwise circular ordering of neighbours around \( w_1 \). This \( z \) might be \( y \) itself, or a neighbour of \( w_1 \) that is closer to \( w_{i-1} \). Now, by planarity, \( z \) is also in \( e \), and if \( N(z) > 1 \), we have in \( z \) a contradiction to the minimality assumption on \( v \), by the same arguments as above. If \( N(z) = 1 \), \( z \) has a legal left support on \( w_{i-1} \), and therefore it can be chosen as \( v_k \), which again contradicts our assumptions.

If \( w_1 = w_{i_j} \), we take \( z \) to be the neighbour of \( w_i \) immediately following \( w_{i+1} \) along the counter-clockwise circular ordering of neighbours around \( w_i \). This \( z \) is in \( e \), and contradicts our assumptions similarly (using a right support for the case \( N(z) = 1 \)).