4 THE AUSLANDER-PARTER ALGORITHM

4.1 Cycle-based planarity testing

Cycle-based planarity testing uses the result of the Jordan Curve Theorem which states that a continuous simple closed curve C divides the rest of the plane into two regions having C as their common boundary. If a point P in one of these regions is joined to a point Q in the other by a continuous curve L in the plane, then L intersects C (Ref 1, p. 66). Also note that by contraposition, if the curve L joining two points P and Q does not intersect with C, then P and Q are in the same region.

Now consider a cycle C in a connected graph G. Then C can be thought of as a simple closed curve (or a Jordan Curve) which separates the plane into two regions; the region enclosed by C and the region outside of C. Now the rest of the connected parts of the graph must lie entirely inside or entirely outside of C. If there exists no such arrangement without edge crossings, then G is non-planar. How this is determined depends on the algorithm used; the Auslander-Parter algorithm is one such algorithm.

4.2 The Auslander-Parter Algorithm

The Auslander-Parter algorithm takes a biconnected graph G as its input and tests it for planarity. Using the result that a graph is planar if and only if all of its bi-connected components are planar (Ref 9, p. 12), the algorithm can be easily adapted to work on graphs which are not necessarily biconnected by performing the algorithm on all of its biconnected components separately.

The Auslander-Parter algorithm takes a cycle C in G, which separates the rest of G with respect to C into k segments, S_i for $i=1..k$ which must lie entirely outside or inside of cycle C (see figure 4.1 and 4.2). Segments can be thought of as the connected components that would be left of G if C were to be removed, including the edges that are incident with the *attachment vertices*; vertices which are common to at least one segment and cycle C, the attachments vertices contained in a segment will also be referred to that segments' *attachments*. Since G is biconnected, each segment contains at least 2 attachment vertices else the removal of an attachment vertex could disconnect a segment from the rest of the graph (implying G is not binconnected). Segments which can be on the same side of C without interleaving are said to be *compatible* with each other and *conflicting* if they can't. A *chord* is a segment which contains only two attachment vertices with an edge (not in C) incident to both of them.

The cycle C is a *separating cycle* if it separates at least two segments; otherwise it is a non-separating cycle.

The three cases shown below summarises the Auslander-Parter algorithm; Examples 4.1 and 4.2 demonstrate the algorithm working on two different graphs. Note that the initial cycle C will have to be found by a separate cycle-finding algorithm such as the one we looked at in section 3.3. In examples 4.1 and 4.2, C has been chosen arbitrarily.

Trivial case: Graph G is a single cycle C. This case can only occur at the beginning of the computation and terminates it.
Base case: Cycle C separates a single segment, which is a path. This terminates the current branch of the
computation (there will be no recursion).
Recursive case: A separating cycle C can be found in G. If the interlacement graph is not bipartite, the algorithm terminates with a non-planarity. Otherwise, recursion is needed on the subgraphs composed by C and each segment.
(Ref 9, p. 14)

Example 4.1

Consider the bi-connected graph G in figure 4.1. If a cycle C is chosen such as the one in figure 4.2, then the cycle separates the rest of the graph into three segments; S_1 , S_2 and S_3 . And so C is a separating cycle so we have the recursive case and an *interlacement graph* is then constructed.

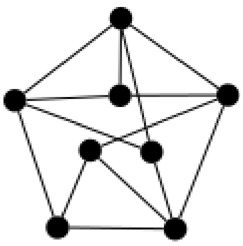


Figure 4.1

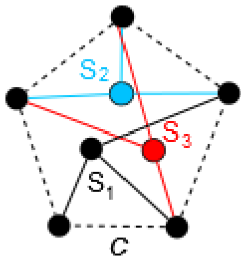


Figure 4.2

In an interlacement graph, the vertices represent segments and edges exist between two vertices if the two corresponding segments are conflicting. If this interlacement graph is not bipartite then the algorithm terminates and G is identified as being non-planar. Otherwise there exists a graph drawing of G in which the k segments do not interlace. The interlacement graph for the three segments in figure 4.2 is shown in figure 4.3 over the next page.

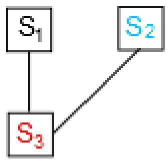
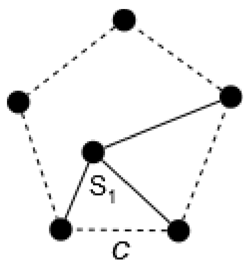
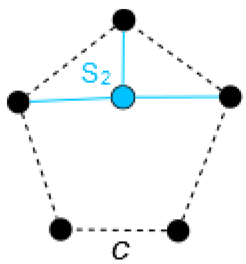


Figure 4.3

The interlacement graph is bipartite and so recursion is needed on the union of each of the three segments with C ; all of which are shown below.





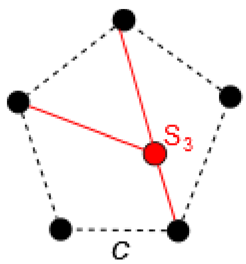


Figure 4.4 **Figure 4.5** **Figure 4.6**

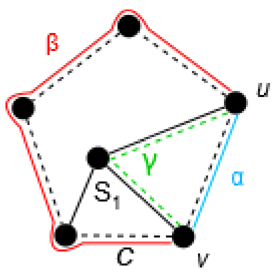
Now, if we consider the graph in figure 4.4 first, looking back at the algorithm we have neither the base case nor the trivial case (since S_1 is not a path). We actually have the recursive case; even though C is not a separating cycle here, a separating cycle can still be found. (Ref 9, p.12)

<u>**Proposition 4.2.1:**</u> If cycle C separates a single segment S which is not a path in a biconnected graph G , then there exists another cycle C' which is a separating cycle.

<u>**Proof:**</u> Cycle C' can be found by choosing two consecutive attachment vertices u and v in the circular ordering of C (see figure 4.7 for an example, where S_1 from figure 4.4 is the single segment which isn't a path). Then let path α be the sub-path of C between u and v containing no other attachment vertices

and let β be the other sub-path of C between u and v. Let path γ be a sub-path of segment S from u to v. Then the cycle C' is $\beta \cup \gamma$ (see figure 4.8 for an example) and α will become a segment which is a path containing attachment vertices u and v. Also since S is not a path and is connected, there exists some edge $e \notin E(C')$ incident to a vertex $w \in \gamma$. Since G is binconnected we know e must therefore be part of a segment containing w as one of its attachment vertices. Hence $S \notin \gamma$ will be part of at least one other segment with respect to C'. Also note that edge e is part of some other segment separate from α and hence so C' separates at least two segments.

□



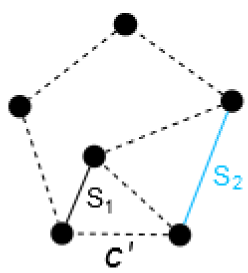


Figure 4.7 **Figure 4.8**

Now a separating cycle has been found, an interlacement graph can be drawn as before. Here, S_1 and S_2 clearly cannot be drawn on the same side of cycle C' and so an edge is drawn between the two vertices representing each segment in the interlacement graph (see figure 4.10).

Figure 4.10

As before, recursion is required on graphs composed of the union of C' and each segment (see figures 4.11 and 4.12). Both graphs have one segment which is a path so we have the trivial case and therefore the algorithm terminates in both cases.

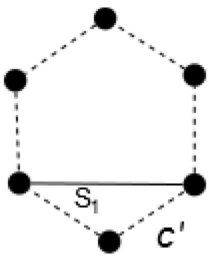


Figure 4.11

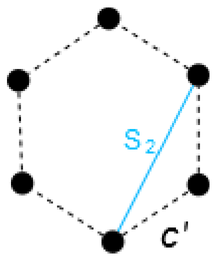
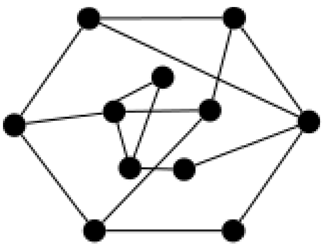


Figure 4.12

After performing the algorithm the same way in the graphs in figures 4.5 and 4.6, you will end up with a similar result. And since the algorithm does not return a non planarity (which can only happen if the interlacement graph is not bipartite), the graph G is planar.

Example 4.2

Now consider the graph H in figure 4.13 below. If a cycle C in H is arbitrarily chosen as in figure 4.14, then it separates the rest of H into two segments S_1 and S_2 as shown.



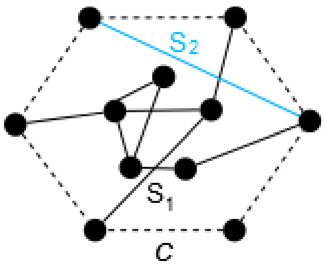
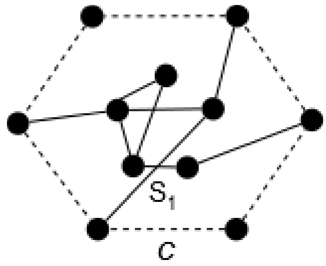


Figure 4.13

Figure 4.14

The interlacement graph for S_1 and S_2 is shown in figure 4.15 below; it is clearly bipartite and so recursion of the algorithm can be carried out on the two graphs $S_1 \cup C$ and $S_1 \cup C$, shown in figures 4.16 and 4.17, respectively.



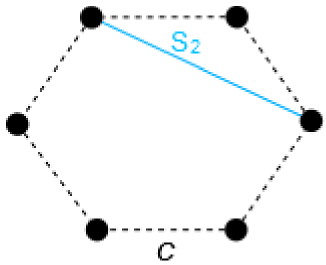
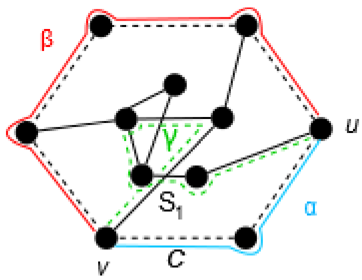


Figure 4.15 **Figure 4.16**
Figure 4.17

Firstly, $S_2 \cup C$ is just a cycle with a path which is the base case and so is planar. Cycle C in the graph $S_1 \cup C$ is not a separating cycle, but the single segment it separates is not a path and so by proposition 4.2.1, a separating cycle can be found in the same way as in Example 4.2. The separating cycle C' in figure 4.19 has been found as in example 4.2; the attachment vertices u and v have been chosen as in figure 4.18 and demonstrate that the two attachment vertices do not have to be adjacent to each other but must still be consecutive attachment vertices with respect to the circular ordering of C .



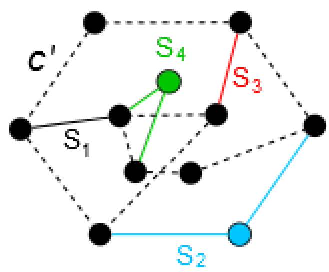


Figure 4.18
Figure 4.19

Figure 4.20 below shows how $S_1 \cup C$ may look if C' were to be 'straightened out' with no edge crossings of itself so as to represent a Jordan Curve, with all the other segments having to lie entirely inside or entirely outside of C'. It is easy to see now that S_1, S_2 and S_3 are all conflicting with each other while S_4 is compatible with every other segment; the corresponding interlacement graph is shown in figure 4.21.

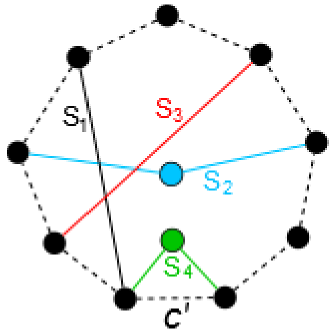


Figure 4.20

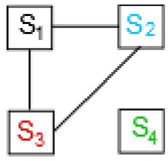


Figure 4.21

The graph is not bipartite since the vertices cannot be divided into two disjoint sets such that there exists no edge incident to two vertices of the same set. Therefore, the algorithm terminates with the result that G is non-planar.

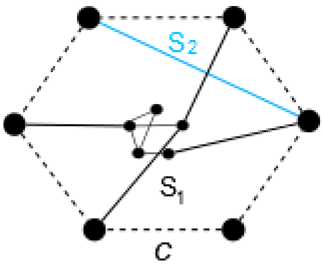
4.3 Analysis of the Auslander-Parter algorithm

In Examples 4.1 and 4.2, it was easy to see which segments were compatible with each other and which segments conflicted without the use of an algorithm. Sometimes it may be too difficult to do by eye or we may wish to implement the Auslander-Parter algorithm as a computer program; in both cases we require an algorithm to test whether segments are compatible or conflicting. When deriving such an algorithm the only information we need about each segment are its attachments.

Let G be a biconnected graph with segments S_i with respect to C .

<u>Proposition 4.3.1</u>: The only property of any two segments $S_i, S_j \in G$ that is required to tell us whether they are compatible or conflicting are their attachments.

<u>Proof</u>: Consider the topological graph of G . Picture what happens if the segments are 'shrunk' (see figures 4.22 and 4.23 below, using the graph from figure 4.14 as an example);



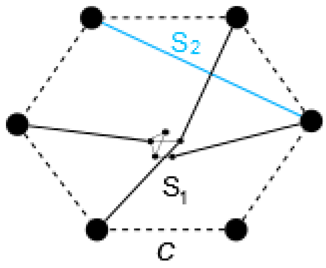


Figure 4.22

Figure 4.23

If the segments are shrunk infinitely then we'd only be left with the edges incident to the attachments. So clearly each edge in S_i that isn't incident to an attachment will not interlace with any other edge in outside of S_i and so we are only left with the edges in S_i which are incident to an attachment. Hence if there is no way to arrange two segments on the same side of C without them interlacing, it is the edges incident to the attachments that interlace; we do not need to consider any other property of the segments.

□

To consider the segments as a whole would likely add needless complexity to the problem of testing if two segments are compatible and so it would make sense to derive a graph transformation which only takes into account segments' attachments.

<u>Definition 4.3.1</u>: Let G be a biconnected graph with cycle C and n segments S_i with respect to C where $a_{i1}, a_{i2}, \ldots, a_{im}$ are the m attachments of each S_i . Then let the transformation $\lambda : G \longrightarrow G^{¿}$ be defined by:

$$\lambda(G) = G^{¿} = C \cup S_1^{¿} \cup S_2^{¿} \cup \ldots \cup S_n^{¿}$$

where $S_i^{¿}=(V,E)=\begin{cases} S_i, \mathit{if}\ S_i\ \mathit{is\ a\ chord} \\ \left(\left\{X_i, a_{i1}, a_{i2}, \ldots, a_{\Im}\right\}, \left\{\left\{X_i, a_{i1}\right\}, \left\{X_i, a_{i2}\right\}, \ldots, \left\{X_i, a_{\Im}\right\}\right\}\right), \mathit{if}\ S_i\ \mathit{is\ not\ a\ chord} \end{cases}$

Less formally, if S_i isn't a chord, then it is represented by a single vertex $X_i \in G^{¿}$ which is adjacent to all the attachments of S_i. If S_i is a chord then modifying the segment in such a way would be unnecessary and so the segment remains the same in $G^{¿}$. Note that cycle C (and its attachment vertices) remains unchanged. To demonstrate this graph transformation, let G be the graph in figure 4.14 of Example 4.2. Then $G^{¿}=\lambda(G)$ is the graph shown in figure 4.24.

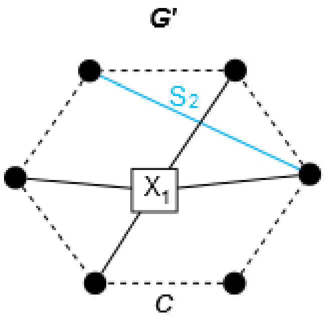


Figure 4.24

Let G be a biconnected graph with segments S_i with respect to C and let $G^{¿}=\lambda(G)$ with segments $S_i^{¿}$ such that $X_i \in S_i^{¿}$ is the only vertex in $S_i^{¿}$ which is not an attachment vertex (see Definition 4.1).

__Lemma 4.3.1__: If the interlacement graph of G is not bipartite then $G^{¿}$ is non-planar.

__Proof__: If the interlacement graph of G is not bipartite then the segments in G cannot be drawn in the plane without some $S_i, S_j \in G$ interlacing. This would mean that $S_i^{¿}, S_j^{¿} \in G^{¿}$ cannot be drawn in the plane without interlacing, implying that $G^{¿}$ is non-planar since $S_i^{¿}$ and $S_j^{¿}$ are just represented by single vertices.

☐

Lemma 4.3.2: Let H be biconnected subgraph of $G^{¿}$. Then there exists a subdivision of H in G.

Proof: Consider H as a union of n paths of length one; $H = P_1 \cup P_2 \cup \ldots \cup P_n$ such that for some $P_j, P_k \subset K$, $(P_j \subseteq S_i^{¿}) \oplus (P_j \subseteq C)$ and no two P_j and P_k share a common edge. There are then three cases to consider; $P_j \subseteq C$, $P_j \subseteq S_i^{¿}$ where $S_i^{¿}$ is a chord and $P_j \subseteq S_i^{¿}$ where $S_i^{¿}$ is not a chord.

If $P_j \subseteq C$ or $P_j \subseteq S_i^{¿}$ where $S_i^{¿}$ is a chord, then there will exist an identical P_j in G since G also contains C and $S_i^{¿}$ (see Definition 4.1). If $P_j \subseteq S_i^{¿}$ and $S_i^{¿}$ is not a chord, then P_j contains an attachment, u, of $S_i^{¿}$ and must also contain vertex X_i, and since H is biconnected, there exists another $P_k \subseteq S_i^{¿}$ such that $P_k \neq P_j$ which also contains X_i and another attachment vertex, v. In G there exists a path $\langle u, s_1, s_2, \ldots, s_m, v \rangle \in S_i$ which is a subdivision of $\langle u, X_i, v \rangle = P_j \cup P_k$ albeit different labelling of vertices. Hence there exists a subdivision of H in G. □

Proposition 4.3.2: A biconnected graph G with cycle C is planar if and only if the following two conditions hold

1. The interlacement graph of G is bipartite
2. Every $S_i \cup C$ is planar for all $S_i \in G$.

Proof:

$(\Rightarrow)$ By way of contradiction, assume that G is planar and either condition 1 or condition 2 is false. If condition 2 is false, then there exists a Kuratowski subgraph in $S_i \cup C$ for some $S_i \subset G$, and since $(S_i \cup C) \subseteq G$, by transitivity G would also contain a Kuratowski subgraph and so G would be non-planar, hence condition 2 must be satisfied if G is planar.

If condition 1 is false then by Lemma 4.3.1, the graph $G^{¿} = \lambda(G)$ is non-planar, implying the existence of a Kuratowski subgraph, K, in $G^{¿}$. Then since Kuratowski subgraphs are biconnected, by Lemma 4.3.2, G must contain a subdivision of K which is still a Kuratowski subgraph and hence G is also non-planar, so condition 1 must be satisfied if G is planar.

$(\Leftarrow)$ If the interlacement graph of G is bipartite, then every $S_i \in G$ with respect to C can be drawn in the plane without interleaving with any other segment, and so clearly, if every $S_i \cup C$ is also planar, then there are no edge crossings between segments and within segments, so G is planar. $\square$