Diploma Thesis
FB Mathematik und Informatik
Freie Universität Berlin

3-Orientations and Schnyder 3-Tree-Decompositions

Construction and Order Structure

by

Enno Brehm

June 2, 2000

Referees:
PD Dr. Stefan Felsner, PD Dr. Klaus Kriegel

Dedicated to my nephew, Luis Christian Jung.

Contents

Introduction

This work is a bit of a patchwork around Schnyder labellings, and is roughly split into three parts: a "purely graph-theoretical" part, an algorithmic part, and the description of an application, whose implementation is part of this thesis.

The first part is concerned with triangular graphs and 3-orientations on them, which are similar to Schnyder labellings and in fact, equivalent to them, as we will see. You can easily move back and forth between them. We will show that given a planar graph, the set of all its 3-orientations forms a distributive lattice, a well-known structure from algebra and order theory. This result was already proved in 1994 by Mendez in [**4**] using bipolar orientations.

To achieve our goal we will investigate on two very basic operations, called the face flip and its "reverse" pendant, the face flop. We will show that in the case of 4-connected triangular graphs these two operations are enough to move between any two 3-orientations,.The face flip imposes a partial ordering on the set of all 3-orientations, but to show that it is a disitributive lattice, we first have to introduce in Chapter 2 another structure, the face potential, which we will show is again equivalent to 3-orientations/Schnyder labellings. We will exhibit the desired lattice structure first in the set of all face potentials and therefrom conclude our initial objective. In Chapter 3 we will transfer our result from 4-connected graphs to general triangular graphs.

Chapter 4 then is concerned specifically with Schnyder labellings and their construction. We will present a variant of an algorithm to construct Schnyder labellings which we think is easier to implement. Moreover we will show how to make this algorithm explicitly construct the maximal and the minimal elements from the distributive lattice we introduced in the first three chapters.

This algorithm has been implemented in a small *LEDA*-based application, which allows you to

- generate and visualize Schnyder labellings and take influence on the generation process,
- create embeddings based on these Schnyder labellings [**8**],
- modify the labellings by applying flips and flops interactively or automatically,

- test graphs on 4-connexity and make graphs 4-connected,
- display the distributive lattice of all Schnyder labellings and its generating poset for 4-connected graphs.

This program is available under `http://www.inf.fu-berlin.de/~brehm/slabel` resp. `/home/sahib/brehm/pub/slabel`. An executable called `slabel` can be found in the subdirectory `bin`. In the subdirectory `src` you will find the sources, including a compilation instruction.

Finally I want to thank Nicole Morawe, Ulli Kortenkamp, Uli Wagner, Alexander Below, Ingo Schurr, Mark Simon for their cooperation and last, but not least, my mom and my brother for their support.

This thesis has been produced using LaTeX and LyX.

CHAPTER 1

3-Orientations

"When shall we three meet again?
In thunder, lightning, or in rain?"

W. Shakespeare
Macbeth

1.1. Basics

Most of the following is basic graph theory as found in any good text-book on this subject, for example [**5**].

$G = (V, E)$ denotes a graph with a set of *nodes* V and a set of *edges* $E \subseteq \binom{V}{2}$. We will be concerned with embedded maximal planar graphs, i.e. if you add another edge to G, then it is no longer planar. This implies that every face is triangular. In the following such graphs will be called *triangular* graphs. The nodes and edges forming the boundary of the exterior face are called *exterior*, all the others *interior*. From Eulers formula we get for any such graph: $|E| = 3|V| - 6$.

A *path* P is a non-repeating sequence $v_1, v_2, \ldots, v_k$, such that v_i and v_{i+1} are adjacent. Its length is $k-1$ and will be denoted by $|P|$. If the edges of the graph are endowed with directions, such a path is called *directed*, if each edge is directed from v_i to v_{i+1}. A *cycle* is a path $v_1, v_2, v_3, ..., v_k$, such that v_1 and v_k are also adjacent. A cycle consisting of three nodes is called a *triangle*. Directed cycles are defined accordingly - and since we are working with embedded graphs, a directed cycle is always *oriented* either clockwise or counterclockwise, also written as *cw-cycle* resp. *ccw-cycle*.

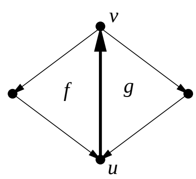


Figure 1.1.1.

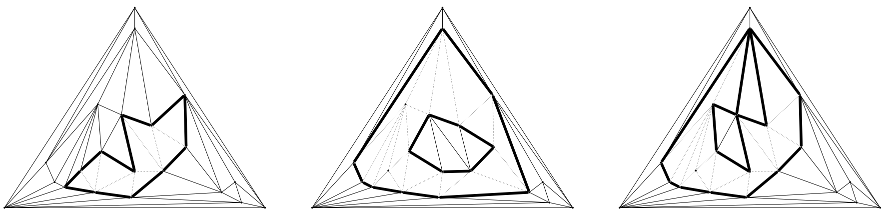
If an edge of an embedded graph is directed, we can speak of its two adjacent faces as the face *left* resp. *right of* it. Or, if two faces f and g are adjacent, then f is called left of g if f is left of the edge that is common to the boundaries of f and g. See Picture 1.1.1.If an edge is directed from u to v, then we say it *leaves* u.

A *region* is an edge-connected set of interior faces of G, where *edge-connected* means that for any two faces f and g of a region R there is a sequence of faces

$$f = f_0, f_1, \ldots, f_k = g$$

such that all of them are in R and f_i and f_{i+1} are adjacent. In terms of the dual graph G^* this can be stated equivalently as: in G^* the set R^* of nodes corresponding to R is connected. A region R is said to be *maximal* with respect to some face property P, if all faces in R have property P and there is no region R' of faces with property P such that R is a proper subset of R'.

Every region has an *outer* boundary, which obviously is a cycle; if there are holes in R, it also has an inner boundary which is formed by one cycle for each hole[1]. If R does not contain holes, it is *simple.*



(a) Simple (b) One hole (c) Two holes

FIGURE 1.1.2. Regions with holes

Although we will heavily use directed edges, for our purposes it makes sense to separate the adjacency information of the graph from the direction (resp. orientation) of the edges because the undirected graphs we are concerned with will not change, but only the direction of the edges. Moreover, we are interested in a special case of oriented edges: a *3-orientation* $\mathcal{T}_G$ of a graph G is obtained by giving every interior edge a direction, such that each interior node v has exactly 3 outgoing edges, i.e. $d^+(v) = 3$, where $d^+(v)$ denotes the *outdegree* of v. There are no constraints for the number of incoming edges, but the above condition already implies that $d^+(v) = 0$ for all exterior nodes v:

PROPOSITION 1.1.1. *In a 3-oriented, triangular graph G no interior edge leaves an exterior node, i.e. interior edges incident to an exterior node are always directed* toward *it.*

[1]For the picky ones: In 1.1.2.c you could argue that it is a simple region or a region with *one* or *two* holes. But in that case you would get repeated vertices on the bounding cycles. So the right interpretation in that case is *two* holes.

PROOF. Let $n = |V|$. Every edge incident to an interior node is an interior one; every interior node has outdegree 3, and there are $n-3$ interior nodes. So we need $3(n-3) = 3n-9$ interior edges to give each node its declared outdegree, but from EULER'S equation we know that there are exactly $3n-6-3 = 3n-9$ interior edges, so there are no edges left to be leaving exterior nodes. (This is no new result, see [8].) □

Note that the same argument will work for any triangle that you might find in a 3-oriented graph. There will be no edge inside the triangle leaving a node on the boundary. In fact, the converse is also true:

PROPOSITION 1.1.2. *If C is a cycle in a 3-oriented triangular graph G and no edge inside C leaves a node on C, then C is a triangle.*

PROOF. This proof is based on the idea used in the proof of Proposition 1.1.1, where we double-count edges. Let $c = |C|$ and let k and l be the number of nodes resp. edges *inside* C. We have

$$l = 3k \tag{1.1.1}$$

because every edge inside C leaves an inner node of C and each one has outdegree 3.

Let now H be the subgraph of G induced by C and its interior. H has $k+c$ nodes and $l+c$ edges. The faces of H are triangular except the outer face which is bounded by c nodes. To triangulate the outer face $c-3$ additional edges are needed, but afterwards H is a triangular graph, and using $|E| = 3|V|-6$ gives

$$\begin{array}{rrcl} & l+c+(c-3) & = & 3(k+c)-6 \\ \Rightarrow & l & = & 3k+c-3 \end{array}$$

However, with (1.1.1) this obviously implies $c = 3$, hence C is a triangle. □

1.2. 3-Tree-Decompositions & Schnyder Labellings

A *3-tree-decomposition* of a triangular graph is a partition of its set of interior edges into three trees, each one being rooted at a different exterior node and containing every interior node, see Fig. 1.2.1(a). In particular, such a partition defines for each interior node v a unique path to each exterior node by walking "toward the root" in each tree. It is easy to see that by directing all edges toward the root of the tree to which they belong, we get a 3-orientation of G: every node is contained in all three trees and therefore has exactly three outgoing edges.

DEFINITION 1.2.1. A *Schnyder labelling* (by Schnyder himself called *realizer*) is a 3-orientation plus a *coloring* of the interior edges: each gets one of three colors (say RED, GREEN, BLUE), such that

1. every interior node has exactly one outgoing edge in each color,
2. the colors of the outgoing edges at every interior node appear always in the same counterclockwise order as seen in Figure 1.2.2(a),
3. incoming edges of one color appear exactly between the outgoing edges of the two remaining colors.

The following fact has been shown by Schnyder in [**8**]:

FACT 1.2.2. *Every triangular graph has a Schnyder labelling.*

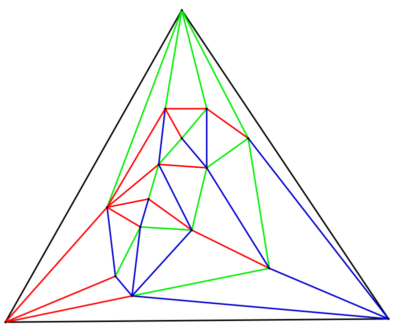
For an example, make the 3-tree-decomposition shown in Fig. 1.2.1(a) to a 3-orientation as described above and you will get a Schnyder labelling. Moreover, Schnyder showed that each Schnyder labelling is also a 3-tree-decomposition — the trees are formed by the set of edges of one color. Together with what has been stated above stated we have:

COROLLARY 1.2.3. *For every triangular graph there exists a 3-orientation and a 3-tree-decomposition.*

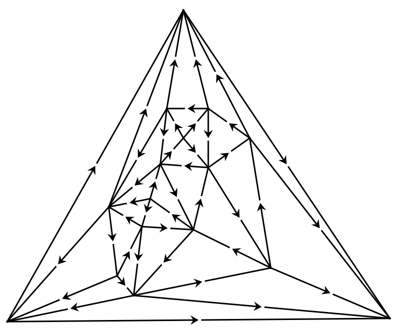
LEMMA 1.2.4. *In a Schnyder labelled triangular graph G, every edge e is the beginning of a unique directed path which ends at an exterior vertex and whose edges are all of the same color as e.*

PROOF. Let v be the node toward which e is directed. There is exactly one edge e_2 of e's color that leaves v. Continue with that edge. Repeat until you reach an exterior vertex (This is possible as the edges of each color form a tree.) □

See Section 4.2 for our version of an algorithm to construct Schnyder labellings. We see that 3-orientations, 3-tree-decompositions and Schnyder labellings are interrelated. Fraysseix/Mendez showed in [**2**]



(a) 3-tree-decomposition



(b) 3-orientation

FIGURE 1.2.1. 3-tree-decomposition and induced 3-orientation

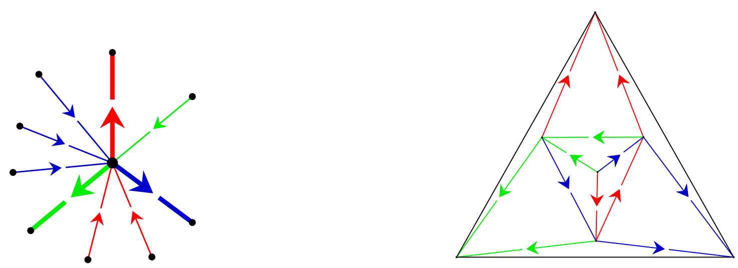
that any 3-orientation of a graph can be extended uniquely (up to "rotation" of the colors) to a Schnyder labelling by choosing the appropriate colors for the edges. So Schnyder labellings are a connecting element between the set of 3-orientations and the set of all 3-tree-decompositions which are induced by Schnyder labellings. This last set is indeed a proper subset of all 3-tree-decompositions, there are 3-tree-decompositions not induced by Schnyder labellings, for an example see Fig. 1.2.2(b). You can also see the 3-orientation that is induced by the 3-tree-decomposition.

In most cases, we will be concerned with 3-orientations, but sometimes it is helpful to know that a 3-orientation can also be viewed as a Schnyder labelling resp. 3-tree-decomposition. Let us note that Schnyder labellings have some nice properties on their own and give relatively compact straight-line-embeddings on the grid (see **[8]**), which are for example used in our application. We will see later that all operations we apply to 3-orientations can also be formulated in such a way that they are immediately applicable to Schnyder labellings, too.

1.3. Cycles in 3-Orientations

In general for a given graph G there exist more than one 3-orientation resp. Schnyder labelling. We will now show an easy way to construct new 3-orientations from one we already have. We shall see that *directed* (or synonymous *oriented*) cycles play an important role here.

PROPOSITION 1.3.1. *Reversing the direction of an oriented cycle in a 3-orientation yields again a 3-orientation.*



(a) Schnyder labelled node

(b) A 3-tree-decomposition that is not Schnyder labelled

FIGURE 1.2.2

PROOF. Let $\mathcal{T}$ be a 3-orientation with an oriented cycle C. Let $\mathcal{T}'$ be obtained from $\mathcal{T}$ by reversing the edges along C. Since there is no change for the nodes not on C, they will still have outdegree 3. So we only have to check that in $\mathcal{T}'$ all nodes *on* C still have 3 outgoing edges. But this is clear because every edge on C has been reversed, so for any node on C one outgoing edge becomes an incoming edge and vice versa, which leaves it again with an outdegree of 3. $\square$

LEMMA 1.3.2. *Let G be a triangular graph. If G has more than one 3-orientation, then every 3-orientation contains directed cycles. More precisely, if $\mathcal{S}$ and $\mathcal{T}$ are two different 3-orientations of G, then there is a cycle C which is clockwise in one of $\mathcal{S}$ or $\mathcal{T}$, and counterclockwise in the other.*

PROOF. Let $\mathcal{S}$ and $\mathcal{T}$ be two different 3-orientations of G. Let e_1 be an edge incident to nodes v_1 and v_2, s.th. e_1 is directed from v_1 to v_2 in $\mathcal{T}$ and from v_2 to v_1 in $\mathcal{S}$. Since e_1 contributes to the outdegree of v_2 in $\mathcal{S}$, but not in $\mathcal{T}$, this deficit must be compensated by some other edge $e_2 \neq e_1$ leaving v_2 in $\mathcal{T}$ but not in $\mathcal{S}$ (If all edges $e \neq e_1$ leaving v_2 in $\mathcal{T}$ were also leaving edges in $\mathcal{S}$, then v_2 would have had an outdegree of 4 in $\mathcal{S}$, which clearly is a contradiction). Let $v_3 \notin \{v_{1,}v_2\}$ be the node that e_2 is directed toward to. We now have a directed path v_1, v_2, v_3 in $\mathcal{T}$ and a directed path v_3, v_2, v_1 in $\mathcal{S}$ and we we can apply the same argument to v_3: e_2 leaves v_3 in $\mathcal{S}$, but not in $\mathcal{T}$, so to get an outdegree of three for v_3 in $\mathcal{T}$, there must some other edge e_3 leave v_2 in $\mathcal{T}$ but not in $\mathcal{S}$, leading us once more to another node v_4, and directed paths v_1, v_2, v_3, v_4 in $\mathcal{T}$ resp. v_4, v_3, v_2, v_1 in $\mathcal{S}$ and so forth.

Eventually we must arrive at a node x_k we have seen before, i.e. such that $x_k = x_i$, for some $1 \leq i < k-1$. But this means that $x_i, x_{i+1}, \ldots, x_{k-1}$ is a directed cycle in $\mathcal{T}$ and $x_{k-1}, \ldots, x_{i+1}, x_i$ is one in $\mathcal{S}$. Hence we have exhibited a cycle which is oriented in opposite ways in $\mathcal{T}$ and $\mathcal{S}$. $\square$

THEOREM 1.3.3. *Let G be a triangular graph. Then there exists at most one 3-orientation without cw-cycles (ccw-cycles).*

PROOF. Suppose that $\mathcal{S}$ and $\mathcal{S}'$ are two different 3-orientations of G which do not contain any cw-cycles, say. By Lemma 1.3.2 there exists a cycle C in both of them, but with different orientations. So in one of them, C must be a cw-cycle, a contradiction. $\square$

1.4. Case Study: Stacked graphs

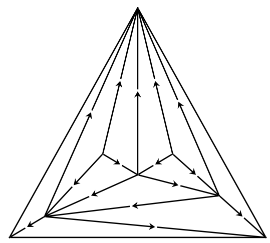
To bring some life to the definitions above we will now introduce a class of stacked graphs for which we will give a different characterization by using 3-orientations.

DEFINITION 1.4.1. Let $G = (V, E)$ be a triangular graph. A node v of G is called a *stacked node*, if $\deg(v) = 3$. G itself is called a *stacked graph* if either

- it consists of three nodes only, or
- G contains at least one stacked node, and $G \setminus v$ is stacked, whenever $v \in V$ is stacked.

Equivalently, a graph is stacked if it can be constructed by starting at a simple triangle and repeatedly putting a new node in a face, connecting it to the nodes on the boundary of that face (FTWK[2]: stacked graphs are the embedded skeleton graphs of a class of 3-polytopes which are known as stacked 3-polytopes.)

An easy induction shows that a stacked graph is always always triangular: a triangle is clearly triangular, and putting a new node into a triangular face will split it into new triangular faces.



Stacked graph

One basic property of stacked graphs is that a 3-orientation can be constructed easily: assume we construct a stacked graph as describe above, by inserting one node at a time. When a node v is inserted into a face f, direct the three edges connecting v to the bounding nodes of f away from v. That way, v will get an outdegree of 3. Since we do this for all interior nodes, in the end we obtain a 3-orientation.

Furthermore, if we have a stacked graph G with a 3-orientation $\mathcal{T}$ and any stacked node v, then all (three) edges incident to v must be directed away from it. It follows that $\mathcal{T}$ induces a 3-orientation for $G \setminus v$: removing v from G will not change the outdegree of any other node. We denote this 3-orientation by $\mathcal{T} \setminus v$. Of course we can repeat this process: the essence is that removing stacked nodes from a 3-oriented graph always leaves a 3-oriented graph and that stacked graphs have a unique 3-orientation. The converse also holds:

PROPOSITION 1.4.2. *A triangular graph $G = (V, E)$ has exactly one 3-orientation if and only if it is stacked.*

PROOF. "$\Leftarrow$" — We will show that a 3-orientation $\mathcal{T}$ of G does not contain any cycles. With Lemma 1.3.2 follows that there can be no two different 3-orientations. So assume G had a directed cycle. Then w.l.o.g. we can assume that one of the nodes on the cycle is stacked (if not, we can remove stacked nodes until at least one of the nodes on the cycle gets stacked; this will leave the cycle completely untouched). But then this stacked node has only outgoing edges, and thus can not be part of a directed cycle, contradiction.

[2]for those who know

"$\Rightarrow$" — By induction on $|V|$: for a graph of 3 nodes there is nothing to show. So let G be a triangular graph with exactly one 3-orientation $\mathcal{T}$ and more than three nodes. There can be no directed cycles in this 3-orientation, because otherwise we could reverse it and get a different 3-orientation (see Proposition 1.3.1). Therefore the graph G together with the 3-orientation $\mathcal{T}$ is a directed acyclic graph[3]. It is well-known that then there must exist a *source*, i.e. a node that has no incoming edges. Let v be this node. Since it has only outgoing edges (and 3 of them), it is a stacked node. If $G \setminus v$ had more than one 3-orientation, each of them could have been extended to G by putting v back in with its edges outgoing. Therefore, $G \setminus v$ itself is a graph with exactly one 3-orientation, thus by induction hypothesis stacked. Hence G must be stacked, too.

□

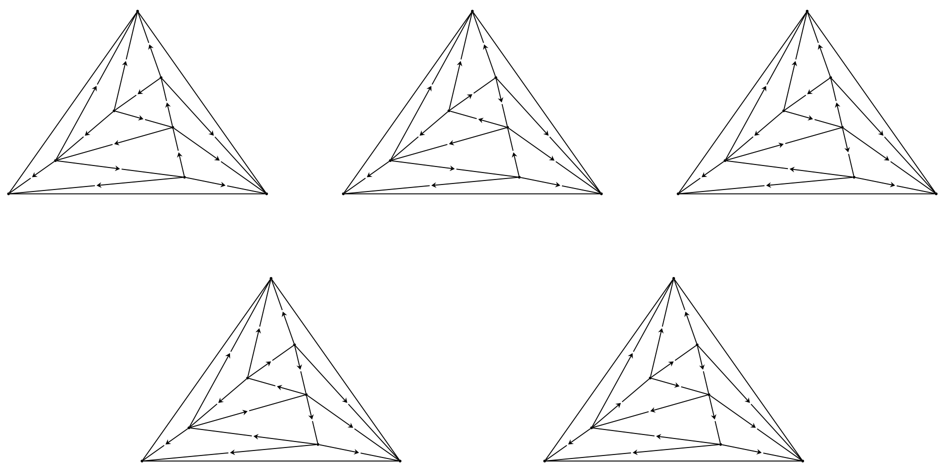


FIGURE 1.4.1. A graph with five 3-orientations

Just to show that there are more interesting graphs than stacked graphs when it comes to 3-orientations, in Fig. 1.4.1 you will find an example of a graph that allows 5 different 3-orientations. You can verify that the first 3-orientation does not contain cw-cycles and that the last one is free of ccw-cycles while the others contain both kind of cycles (see Theorem 1.3.3).

1.5. Cycles and Triangles

As we have already seen, reversing cycles plays an important role in getting from one 3-orientation to another. To get a deeper understanding of this, we will now restrict ourselves to the smallest possible change

[3]We can safely neglect the exterior edges as they are not directed.

that we can make this way: if we allow only triangles to be reversed. In the following sections we will show that nothing is lost by this restriction. We will first show that an oriented cycle always comes with an oriented triangle:

LEMMA 1.5.1. *Be G a triangular graph with a 3-orientation $\mathcal{T}$. If $\mathcal{T}$ contains a directed cycle, then it also contains a triangle with the same direction.*

PROOF. W.l.o.g. let C be a ccw-cycle in $\mathcal{T}$ containing the least number of faces. We will show that no edge inside C is leaving a node on C, so by Proposition 1.1.2 C is a triangle.

Assume a node v on C has such an edge e. In Section 1.2 we saw that $\mathcal{T}$ can also be considered as a 3-tree-decomposition: Lemma 1.2.4 then allows us to view e as the beginning of a directed path from v to an exterior node x, which obviously lies outside C . Hence, the directed path from v to x must cross C and eventually leave it at another node w, yielding a directed path from v to w, see Fig. 1.5.1. But such a path cannot exist, since it formed a directed "chordal" path, which would immediately allow us to find a cycle containing less faces than C. (In Fig. 1.5.1 P and C' together are a ccw-cycle.) ☐

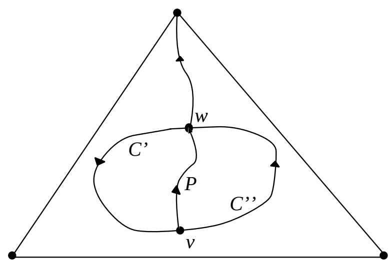


FIGURE 1.5.1. A "chordal" path crossing C

Given an oriented cycle C consisting of more than three nodes, this proof also shows how to find an identically oriented triangle: start at a node on the boundary of C that has an outgoing edge pointing into the interior of C. Follow the according path to an exterior node until it leaves C; this yields a new cycle C_o that contains fewer faces than C. Repeat the process until you no longer find any edges pointing inwards. At this point the cycle is a triangle. It follows immediately from this construction that this triangle is *contained* in the original cycle C, which yields the following corollary.

COROLLARY 1.5.2. *Given an oriented cycle C in a 3-orientation $\mathcal{T}$, then C is a triangle or it contains a triangle with the same orientation as C.*

1.6. 4-Connected Graphs

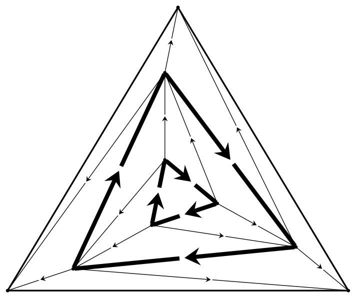


FIGURE 1.6.1. Nested triangles

The two cases in Corollary 1.5.2 are in general not mutually exclusive: a triangle can very well contain another one, for example see Fig. 1.6.1. This leads us to a class of graphs in which these two cases do exclude each other: i.e. graphs in which *every* triangle is the boundary of a face.

In some sense, such a graph is the opposite of a stacked graph, because in the latter (if it is not the simple triangle), you can *always* find a triangle that is *not* a face boundary: namely, the nodes adjacent to a stacked node always form a triangle.

DEFINITION 1.6.1. A connected graph G is 4-*connected* if it cannot be disconnected by the removal of less than 4 nodes.

While this is the standard definition, for the case of triangular graphs another characterization will prove more useful.

DEFINITION 1.6.2. A triangle t of a triangular graph G is a *separating triangle*, if it is not the boundary of a face, in other words, if there there are nodes inside and outside of t.[4]

For an example of a separating triangle take another look at Fig. 1.6.1: the outer bold triangle is a separating triangle.

PROPOSITION 1.6.3. *A triangular graph G is 4-connected if and only if it does not contain separating triangles.*

We will not prove thie here, let us just mention that the "only if" direction is easy: after removal of a separating triangle G breaks down into the the parts inside and outside the separating triangle. For the "if" direction, see [**1**].[5]

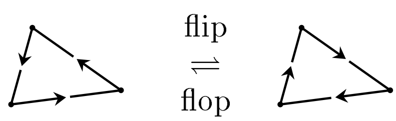
[4]Instead of separating triangles one can also speak of nontrivial triangular subgraphs: separating triangles are just the outer face-boundaries of these subgraphs.

[5]It is interesting to note that in this work you will also find a linear time algorithm to check 4-connexity of a planar graph using 3-orientations.

Proposition 1.6.3 states that in 4-connected graphs triangles are always face boundaries. This means that any triangle can be identified with a face. Together, 4-connected graphs and the restriction to triangles lead to a shift in concept: from the the reversal of directed cycles toward *flipping faces.* In Section 3.2 we will transfer our results back to general triangular graphs. But let us see now how far we can get with face flips on 4-connected graphs.

1.7. Faces & Flips

DEFINITION 1.7.1. Let G be a 4-connected triangular graph. A face f of G whose boundary does not contain an exterior edge is called *flippable* with respect to a 3-orientation $\mathcal{T}$ if it is on the left side of all its bounding edges. Analogously, it is called *floppable* if its boundary edges are oriented in such a way that f is right of all of them. In other terms, the boundary of a flippable face is a ccw-triangle and that of a floppable face is a cw-triangle.



The *face flip* (or just *flip*) will from now on be the basic operation to obtain new 3-orientations from existing ones: take a flippable (floppable) face f and reverse the direction of the bounding edges. If $\mathcal{T}$ is a 3-orientation and f is a flippable (floppable) face in $\mathcal{T}$, then let $\mathcal{T}_f$ denote the 3-orientation obtained by flipping (flopping) f. Since a face can obviously never be both flippable *and* floppable at the same time, this notation is unambigous. It is clear that if f is flippable in $\mathcal{T}$, then f is floppable in $\mathcal{T}_f$ and vice versa.

DEFINITION 1.7.2. A sequence of faces $(f_1, f_2, f_3, ...)$ of a graph G with a 3-orientation $\mathcal{T}$ is called a *flip sequence on* $\mathcal{T}$, if f_1 is flippable in $\mathcal{T}$, f_2 is flippable in $\mathcal{T}_{f_1}$, f_3 in $\mathcal{T}_{f_1 f_2}$ and so on. A *flop sequence* is analogously defined. If S is a flip (flop) sequence, we will write $\mathcal{T}_S$ for the 3-orientation obtained by flipping all faces in the order in which they appear in S. The length of the sequence will be denoted by $|S|$. Furthermore, a flip sequence S is *maximal* if there exists no flippable face in $\mathcal{T}_S$ (analogously for flop sequences). Note that by definition these sequences contain either **only** flippable **or** floppable faces, i.e. they are not "mixed". If S is a flip (flop) sequence on $\mathcal{T}$ and S' is a flip (flop) sequence on $\mathcal{T}_S$, then S and S' can be concatenated to form a single flip (flop) sequence on $\mathcal{T}$, written SS'.

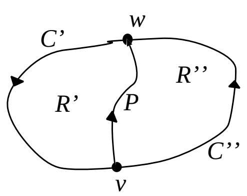
It is obvious that if S is a flip sequence on $\mathcal{T}$, then the reversal of S is a flop sequence on $\mathcal{T}_S$, and vice versa. Of course this makes sense only if S is finite, but we will soon see that this is always the case.

We will now see that in the case of 4-connected graphs we can easily simulate the reversal of a directed cycle by a sequence of flips (flops) of the faces contained in the cycle.

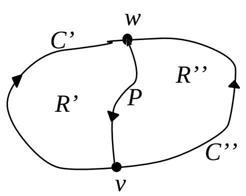
PROPOSITION 1.7.3. *Let G be a 4-connected, triangular graph with a 3-orientation $\mathcal{T}$ and let C be a ccw-cycle in $\mathcal{T}$. Let c denote the number of faces inside C. Then there is a flip sequence S of length c, in which every face inside C appears exactly once, s.th. $\mathcal{T}_S$ is identical to $\mathcal{T}$ except that C is a cw-cycle in $\mathcal{T}$. That is, S reverses C. (A symmetric statement for cw-cycles is proved accordingly.)*

PROOF. By induction on k: if $c = 1$ then C bounds a single flippable face f and $S = (f)$ will do. If C contains more than one face, then C must consist of more than three nodes (this is because G is 4-connected: otherwise C would be a separating triangle).

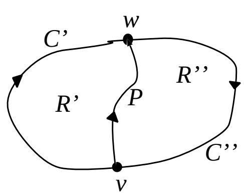
Due to Proposition 1.1.2 we know that there is at least one edge inside C that leaves some node v on C. Using the same reasoning as in the proof of 1.5.1 we can extend this edge to a directed path crossing the interior of C between v and another node w (Figure 1.7.1.a). This will split the region bounded by C into two smaller regions R' and R'': one is bounded by C' and P, the other one by P and C''.



(a) $\mathcal{T}$



(b) $\mathcal{T}_S$



(c) $\mathcal{T}_{SS'}$

FIGURE 1.7.1. Reversing a cycle

C' and P together together form a ccw-cycle that contains less faces than C. So by induction hypothesis there is a flip sequence S, s.th. in $\mathcal{T}_S$ the cycle formed by C' and P is now clockwise (Figure 1.7.1.b). Since this implies that the direction of P is reversed, in $\mathcal{T}_S$ now C'' and P form a ccw-cycle. Using the induction hypothesis again (this time on $\mathcal{T}_S$) we get a second flipsequence S', s.th. in $\mathcal{T}_{SS'}$ C'' and P now form a clockwise cycle(Figure 1.7.1.c). In particular, the edges on P will again have their original orientation. So $\mathcal{T}_{SS'}$ and $\mathcal{T}$ differ only in the direction of C and SS' is the desired flip sequence. It is clear that every face in C appears exactly once either in S or S', which also implies $|SS'| = c$. $\square$

Given a 3-orientation we are interested in how far we can get by flipping or flopping faces: which 3-orientations can we obtain this way? (Remember that if a graph has more than one 3-orientation, then any 3-orientation contains directed cycles and thus flippable or floppable faces.) One central result will be that we can get *all* 3-orientations this way, no matter where we start.

Loops are another issue we are concerned with: while it is clear that we might run into a loop if we use flips *and* flops, we will now see that we can not loop as long as we do only flips *or* flops but not both. To show this we will bound the length of flip sequences by estimating the overall number of times a face can be flipped. Much of the rest of this chapter is based on the following easy, but fundamental, observation:

OBSERVATION 1.7.4. *(Interleaving flips) Let f be a face in a 4-connected, triangular graph G. Then in any flip sequence S, between two flips of f ,there must occur a flip of each neighbour of f (this does not mean that these flips must occur immediately following each other, there might be other face-flips between them.)*

PROOF. To be flippable, a face f needs to be left of all its neighbours. If f is right of an adjacent face g, the only way to make f be left of is to actually flip g. So, if f gets flipped, all its neighbours must be flipped before f itself can be flipped again. □

LEMMA 1.7.5. *In a 4-connected, triangular graph G the maximal number of times a face can be flipped in a 3-orientation is bounded.*

PROOF. It should be clear that those faces which are not bounded by interior faces only cannot be flipped at all because exterior edges are not directed. This makes zero a perfect bound for those faces. Together with Proposition 1.7.4 — the interleaving of flips — this already yields a bound for *all* faces: a face f adjacent to a face g with a zero bound can be flipped at most once, because a second flip on f would make a flip on g necessary; in general, if we know that a face f can be flipped at most x times, then all its neighbours can be flipped at most $x + 1$ times, and will laready bound the number of flips on any face.[6] □

COROLLARY 1.7.6. *The length of any flip sequence is bounded.*

PROOF. The number of flips for any face is bounded, so it can only appear a finite number of times in a flip sequence. Hence a flip sequence cannot get arbitrarily long. □

[6]Obviously the smallest "distance" of a face f to any zero-bound face gives an upper bound for it. This can be formalized by using the dual graph G^* and declaring this distance to be the length of the shortest path from f^* to any g^*, s.th. g has a zero bound. The proof itself is then an easy induction on the distance.

This make it easy to show that there indeed exist 3-orientations that have only one type of oriented cycles. (Comp. Theorem 1.3.3)

DEFINITION/LEMMA 1.7.7. *Let G be a 4-connected triangular graph. Then there exist unique 3-orientations without ccw-cycles resp. cw-cycles. We will denote these 3-orientations by $\mathcal{L}_G$ resp. $\mathcal{R}_G$.*

PROOF. The uniqueness part has been done in Proposition 1.3.3. We will do the existence part only for $\mathcal{L}_G$, it is analogous for $\mathcal{R}$. Let $\mathcal{T}$ be a 3-orientation. We have seen that the length of a flip sequence is bounded, so we can not infinitely flip faces. Accordingly, we must eventually reach a 3-orientation $\mathcal{T}'$ in which no face is flippable. Then it follows from Lemma 1.5.1 that $\mathcal{T}'$ does not contain ccw-cycles at all, thus it is $\mathcal{L}_G$. □

See also Section 4.2 where an algorithm is presented that can explicitly construct these two 3-orientations.

COROLLARY 1.7.8. *Let G be a 4-connected triangular graph. Two 3-orientations are* neighbors *if one is obtained from the other by a single face flip/flop. Then the graph of all possible 3-orientations of G together with this neighbor relation is connected.*

PROOF. Take any two 3-orientations $\mathcal{T}$ and $\mathcal{T}'$. Let S and S' be maximal flip sequences on $\mathcal{T}$ resp. $\mathcal{T}'$. Then clearly $\mathcal{T}_S = \mathcal{T}'_{S'} = \mathcal{L}_G$. So first applying all flips from S to $\mathcal{T}$ and then flopping the faces from S' in reverse order connects $\mathcal{T}$ and $\mathcal{T}'$. □

In the light of Corollary 1.7.6 the following definitions make sense:

DEFINITION 1.7.9. For a 3-orientation $\mathcal{T}$ on a 4-connected triangular graph G, define $\textsc{rank}(\mathcal{T})$ to be the length of a *shortest* maximal flip sequence. For a face f define the *rank of* f, denoted by $\textsc{rank}^{\mathcal{T}}(f)$, to be the *largest* number of times f appears in a maximal flip sequence.

The following are some basic properties following from these definitions.

LEMMA 1.7.10. *Let G, $\mathcal{T}$ be defined as above and let f and g be two adjacent faces. Then the following holds:*

1. $\textsc{rank}(\mathcal{T}) \leq \sum_f \textsc{rank}^{\mathcal{T}}(f)$
2. $|\textsc{rank}^{\mathcal{T}}(f) - \textsc{rank}^{\mathcal{T}}(g)| \leq 1$
3. $\textsc{rank}^{\mathcal{T}}(f) \geq \textsc{rank}^{\mathcal{T}}(g)$, *if f is left of g*

PROOF. (1) Let S be a shortest maximal flip sequence on $\mathcal{T}$. For any face f let s_f denote how often f appears in S. Then clearly $s_f \leq \textsc{rank}^{\mathcal{T}}(f)$, hence $\textsc{rank}(\mathcal{T}) = |S| = \sum_f s_f \leq \sum_f \textsc{rank}^{\mathcal{T}}(f)$.

(2) This is a direct consequence from the interleaving of flips: if we have a maximal flip sequence with f appearing k times, then g must

appear at least $k-1$ times, which implies $\textsc{rank}^{\mathcal{T}}(g) \geq \textsc{rank}^{\mathcal{T}}(f) - 1$. By reasons of symmetry we also get $\textsc{rank}^{\mathcal{T}}(f) \geq \textsc{rank}^{\mathcal{T}}(g) - 1$, which together yields $|\,\textsc{rank}^{\mathcal{T}}(f) - \textsc{rank}^{\mathcal{T}}(g)| \leq 1$.

(3) If f is left of g, then in a maximal flip sequence with g appearing $k > 0$ times there are $k-1$ flips of f interleaved between the flips of g and one additional flip of f must occur *before* the first flip of g to make g flippable at all. $\square$

It might seem that we introduce a bias towards *flips* here, but by the symmetry of flips and flops the results immediately carry over to flops.

CHAPTER 2

Face potentials

"Two wrongs don't make a right, but three lefts do."

ANONYMOUS

Lemma 1.7.5 and 1.3.3 together show that in a 4-connected triangular graph G, the structure of all 3-orientations together with the face-flip operation is well-behaved: there are no non-terminating sequences, and there are two special 3-orientations reachable from any 3-orientation by flipping resp. flopping faces. However, there is even more structure, but to show this we need to examine more carefully of how often a face can actually be flipped. We will see that combining the idea from Lemma 1.7.5 with 3-orientations leads us to tight bounds.

2.1. Maximal, Consistent and Induced Potentials

DEFINITION 2.1.1. A *potential* X of a 4-connected, triangular graph G is a mapping $f \mapsto x_f$ from the interior faces to the natural numbers, such that

- $x_f = 0$, if the boundary of f contains an exterior edge
- $|x_f - x_g| \leq 1$ holds for two adjacent faces f, g

The *value* of a potential X, denoted by $|X|$, is defined by

$$|X| := \sum_f x_f.$$

For any face f, x_f is also called its (potential) value or just its potential. If an edge e is between two faces with different values, then we speak of a *potential difference at* e and call e a *separating* edge.

The faces whose boundary contains an exterior edge are called *fixed faces* because they are always valued zero in any potential, see Fig. 2.1.1.

Obviously, setting $x_f = 0$ for every face f always yields a potential, so this concept alone will not help in describing the rank of a face. Therefore let us further exploit the idea used in the proof of Lemma 1.7.5. The interleaving of flips implies that if a face f can be flipped k times, then all adjacent faces can be flipped at most $k + 1$ times This follows from the incidence structure of G alone. However, similar to Lemma 1.7.10 we can take into account a concrete 3-orientation $\mathcal{T}$ to gain more information: if f is left of g in $\mathcal{T}$, then *if* one of them can be possibly flipped more often, it must be f. This is because the face on the left side will get flippable first, and afterwards they will stay "in sync".

The interleaving of flips as well imposes that in this case the number of flips applied to f will always be greater or equal to the number of flips applied to g. The next definition formalizes this, by calling a potential X *consistent with a 3-orientation* $\mathcal{T}$, if it respects this fact, i.e. whenever two adjacent faces differ in X then the larger value must be on the left side of the separating edge.

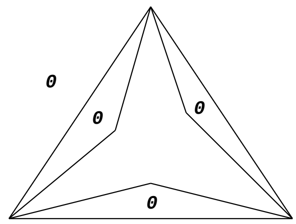


FIGURE 2.1.1. Fixed faces of G

DEFINITION 2.1.2. A potential X of a graph G is *consistent* with a 3-orientation $\mathcal{T}$ if for any directed edge e — let f and g be the faces left resp. right of it — $x_f \geq x_g$ holds. Together with the above definition of a potential we get

$$x_g \leq x_f \leq x_g + 1.$$

Moreover, X is called *maximal consistent* with $\mathcal{T}$ if for all potentials Y consistent with $\mathcal{T}$

$$|Y| \leq |X|$$

holds.

The important point in this definition is that consistency is a *local* condition — we can check it edge by edge. So if we are given a 3-orientation $\mathcal{T}$ and a potential X consistent with it and want to manipulate X to get another consistent potential or manipulate $\mathcal{T}$ by reversing some edges, we will only have to check edges at which a change actually occurs.

In most cases we do not even have to check all of them. For example, if we construct X' from X by taking a region R and increase all potential values inside this region by, say, one, then we only need to check the edges *along the boundary* of R to verify that X' is again consistent with $\mathcal{T}$: outside this region nothing changed, and inside R all faces changed by a constant value, which obviously will not invalidate the inequalities.

On the other hand, we could for example safely reverse a directed cycle in $\mathcal{T}$ as long as it does not contain a separating edge: X would still be consistent.

The following lemma immediately shows that we have already seen a consistent potential:

LEMMA 2.1.3. *Let G be a 4-connected triangular graph and $\mathcal{T}$ a 3-orientation of G. Then*

$$X\colon f \mapsto \mathrm{RANK}^{\mathcal{T}}(f)$$

is a potential consistent with $\mathcal{T}$.

PROOF. follows immediately from Lemma 1.7.10. □

DEFINITION/LEMMA 2.1.4. *Let $\mathcal{T}$ be a 3-orientation of a graph G, and let X, Y be two potentials consistent with $\mathcal{T}$. Define Z face-wise by*

$$f \mapsto z_f := \max{(x_f, y_f)}.$$

Then Z again is a potential, consistent with $\mathcal{T}$. By a slight abuse of notation, we also denote this potential by $\mathrm{MAX}(X,Y)$. $\mathrm{MIN}(X,Y)$ *is defined accordingly and it also is a potentials consistent with $\mathcal{T}$.*

PROOF. Fixed faces clearly get mapped to zero by Z. Let now f and g be two adjacent faces, such that f is left of g. Then we have

$$x_g \leq x_f \leq x_g + 1 \quad \text{and} \quad y_g \leq y_f \leq y_g + 1$$

which implies

$$\begin{aligned} \max(x_g, y_g) &\leq \max(x_f, y_f) \\ &\leq \max(x_g + 1, y_g + 1) = \max(x_g, y_g) + 1. \end{aligned}$$

However, this is just $z_g \leq z_f \leq z_g + 1$. The proof for $\mathrm{MIN}(X,Y)$ is analogous. □

COROLLARY 2.1.5. *For a graph G with a 3-orientation $\mathcal{T}$, the maximal consistent potential is unique.*

PROOF. Assume X and Y are two different maximal consistent potentials. Necessarily we have $|X| = |Y|$. Since $X \neq Y$, there exist w.l.o.g. a face f s.th. $x_f > y_f$. But this implies $|\mathrm{MAX}(X,Y)| > |Y| = |X|$, contradicting the assumption that X and Y are maximal valued consistent potentials. □

This implies that we could also define the unique maximal potential face-wise by

$$f \mapsto \max\{x_f : X \text{ consistent potential}\}\ .$$

Now that this potential is well-defined, the following definition is reasonable.

DEFINITION 2.1.6. A potential is said to be *induced* by a 3-orientation $\mathcal{T}$ if it is maximal consistent with it.

See Fig 2.1.2 for an example of a induced potential. Our initial objective was to give a better bound on the rank of a face in a 3-orientation $\mathcal{T}$. From Corollary 2.1.5 and Lemma 2.1.3 immediately follows that $\textsc{rank}^{\mathcal{T}}(f) \leq x_f$ holds for every face f if x_f is its value in the potential induced by $\mathcal{T}$. It will take a little more work to show that in fact both numbers are equal.

2.2. Consistent Potentials & Directed Cycles

Lemma 2.2.1. *Let G be a 4-connected, triangular graph with a 3-orientation $\mathcal{T}$ and let X be a potential consistent with $\mathcal{T}$. If in $\mathcal{T}$ we have a*

1. *ccw-cycle C that does not contain separating edges or*
2. *a cw-cycle C consisting only of separating edges,*

then X is not maximal.

Proof. The idea is to increase the potential value of each face inside C by one. It is not hard to see that this will again yield a potential consistent with X: we will only have to check the edges that form C. In the first case, all these edges are now separating edges - but since C is a ccw-cycle, the higher value (inside C) will always be on the left side and by construction, the difference is exactly one. In the second case, because C is a clockwise cycle, before we raise the interior of C the greater value is always on the outside - so by raising its interior the potential differences along C will disappear and no new one will be introduced. □

One of our goals is to map 3-orientations one-on-one to the induced potentials. The next lemma will get us closer.

Lemma 2.2.2. *A potential X can be be induced by at most one 3-orientation.*

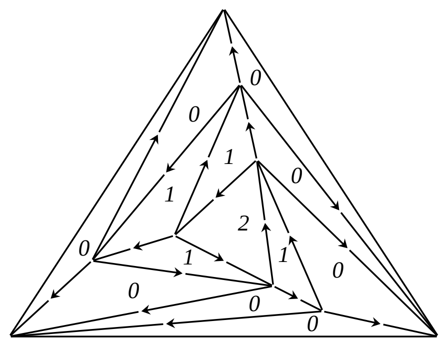
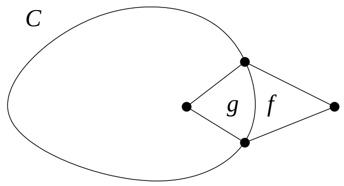


Figure 2.1.2. Induced potential



PROOF. Let $\mathcal{T}$ and $\mathcal{S}$ be two 3-orientations such that X is maximal consistent with both of them. If $\mathcal{S} \neq \mathcal{T}$, then from Lemma 1.3.2 we know that there is a cycle C which w.l.o.g. is oriented clockwise in $\mathcal{S}$ and counterclockwise in $\mathcal{T}$. Let e be an edge of C, and f and g its adjacent faces, with f outside and g inside C, see figure above.

In $\mathcal{T}$ we find g to the left of f which implies $x_f \leq x_g$ while in $\mathcal{S}$ f is left of g and that means $x_g \leq x_f$. Hence $x_f = x_g$, and since this holds for all edges e of C, it is a ccw-cycle in $\mathcal{T}$ along which no potential difference exists, hence by Lemma 2.2.1 it cannot be maximal consistent with $\mathcal{T}$, contradiction. $\square$

2.2.1. Heightcycles. There is another aspect of potentials. We can interpret them as a kind of height-functions, putting each face on a certain *region*. Then the standard definition for potentials states that two adjacent faces can only differ in height by one, so there are no steps greater than one. This introduces the idea of a *height*cycle: a cycle formed by edges along which the potential changes from some value $k-1$ to k.

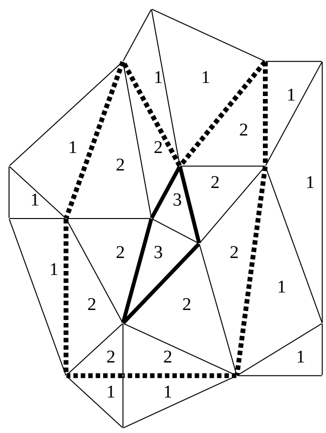


FIGURE 2.2.1. k-regions

DEFINITION 2.2.3. Let G be a triangular, 4-connected graph together with a potential X and $k > 0$. A *k-region* is a maximal region R such that

$$x_f \geq k, \text{ for all } f \in R.$$

See Figure 2.2.1 for an example of a 2-region and a 3-region. Note that the 3-region is completely contained in the 2-region.

Clearly, a k-region R must be bounded by faces with value less than k, therefore the boundary of R consists of edges where faces with value k — inside R – and $k-1$ — outside R – meet. So the mentioned *heightcycles* are just the boundaries of k-regions. It is easy to see that every separating edge is part of exactly one heightcycle and every face with potential value k is contained in exactly one k-region.

LEMMA 2.2.4. *Let G be triangular and 4-connected. Let $\mathcal{T}$ be a 3-orientation of G and X be a potential consistent with $\mathcal{T}$. Moreover let R be a k-region of X, $k > 0$. Then the outer boundary of R is a ccw-cycle in $\mathcal{T}$ and any inner boundary of R is a cw-cycle.*

PROOF. We will show the lemma only for the outer boundary, so let e be an edge of the outer boundary of R. We know e separates a face inside R with value k from a face outside R with value $k-1$. Since X is consistent with $\mathcal{T}$, the inner face must be left of the outer face, and this holds for all edges along the outer boundary. So this outer boundary of R forms a counterclockwise cycle. Applying the same argument to the edges of the inner boundary yields the desired result. □

DEFINITION 2.2.5. Let G be a triangular, 4-connected graph and X be a potential of G. X is called *monotonous*, if no k-region contains a hole, for all values of k.

Intuitively, this means that a monotonous potential does not have "valleys", that is regions that are completely sourrounded by faces with higher value. We will see that induced potentials have this property which makes life a bit simpler in some cases.

LEMMA 2.2.6. *Every induced potential is monotonous.*

PROOF. Let G be a triangular 4-connected graph with a 3-orientation $\mathcal{T}$ and let X be the potential induced by $\mathcal{T}$. Assume there exists a $k > 0$ and a k-region R with a hole. Then the boundary of the hole is an inner boundary of R and thus a cw-cycle, consisting of separating edges only. Lemma 2.2.1 implies that X is not maximal, a contradiction. □

Obviously, the converse of Lemma 2.2.6 is not true because decreasing all values greater zero in a monotonous potential by one will again yield a monotonous potential which is consistent but not maximal, not to mention the zero potential.

PROPOSITION 2.2.7. *Let G be a 4-connected triangular graph. Let X be the potential induced by $\mathcal{L}_G$. Then X is zero on all faces.*

PROOF. We know $\mathcal{L}_G$ is ccw-cycle-free. Assume X is not constantly zero. Then we find a face f with a potential value $k := x_f > 0$, and a k-region R containing f. However, the outer boundary of a k-region is always a ccw-cycle (Lemma 2.2.4), contradiction. So X must be zero on all faces, so this potential is maximal consistent with $\mathcal{L}_G$. $\square$

The converse also holds since we have seen that a potential can be induced by at most one 3-orientation (Lemma 2.2.2.) So we have found our first concrete pair of 3-orientation with induced potential — and this induced potential describes exact for each face in G how many flips will be possible on it in $\mathcal{L}_G$: zero.

2.3. Potentials and Flips

We will now examine how the induced potential changes under face flips.

DEFINITION 2.3.1. Given a mapping $X : f \mapsto x_f \in \mathbb{N}$ and a face g, define a mapping $X^{[g-]}$ as follows

$$x_f^{[g-]} := \begin{cases} x_f - 1, & \text{if } f = g \\ x_f & \text{otherwise} \end{cases}$$

Analogously define $X^{[g+]}$ to be identical to X, but with the value of g increased by one.

Obviously, $X^{[g+][g-]} = X^{[g-][g+]} = X$. Note that if X is a potential, this does not imply that $X^{[g+]}$ and $X^{[g-]}$ are potentials. If they are, then $|X^{[g+]}| = |X| + 1$ and $|X^{[g-]}| = |X| - 1$.

PROPOSITION 2.3.2. *Be G a triangular 4-connected graph with a 3-orientation $\mathcal{T}$. Furthermore let X be a potential consistent with $\mathcal{T}$ and f be a face. Then*

1. if f is flippable, then $X^{[f-]}$ is a potential consistent with $\mathcal{T}_f$,
2. if f is floppable, then $X^{[f+]}$ is a potential consistent with $\mathcal{T}_f$,
3. in both cases, if X is induced by $\mathcal{T}$, then $X^{[f-]}$ resp. $X^{[f+]}$ will be induced by $\mathcal{T}_f$.

PROOF. (1) Let f be flippable. We only need to check the edges bounding f to see whether $X^{[f-]}$ is a potential consistent with $\mathcal{T}_f$ because there is no change at any other edge. Therefore let g be a face adjacent to f. Since f is flippable in $\mathcal{T}$, g must be right of it. This gives:

$$\begin{aligned} x_g \le x_f \le x_g + 1 \quad &\Leftrightarrow \quad x_g^{[f-]} \le x_f^{[f-]} + 1 \le x_g^{[f-]} + 1 \\ &\Leftrightarrow \quad x_f^{[f-]} \le x_g^{[f-]} \le x_f^{[f-]} + 1 \end{aligned}$$

so $X^{[f-]}$ is consistent with $\mathcal{T}_f$, in which f is now floppable, so g is left of f.

(2) Like (1), but now f is floppable, so any g adjacent to f is left of it. Thus,

$$\begin{aligned} x_f \leq x_g \leq x_f + 1 \quad &\Leftrightarrow \quad x_f^{[f+]} - 1 \leq x_g^{[f+]} \leq x_f^{[f+]} \\ &\Leftrightarrow \quad x_f^{[f+]} \leq x_g^{[f+]} + 1 \leq x_f^{[f+]} + 1 \\ &\Leftrightarrow \quad x_g^{[f+]} \leq x_f^{[f+]} \leq x_g^{[f+]} + 1 \end{aligned}$$

and again, the constructed potential is consistent with $\mathcal{T}_f$, in which f is flippable in this case.

(3) We will show this only for the case that f is flippable, the other one is analogous. So let f be *flippable* and X be induced by $\mathcal{T}$: assume there is a potential Z, consistent with $\mathcal{T}_f$, s.th. $|Z| > |X^{[f-]}|$. Moreover f is floppable in $\mathcal{T}_f$. So, by (2) $Z^{[f+]}$ is consistent with $(\mathcal{T}_f)_f = \mathcal{T}$. However,

$$|Z^{[f+]}| = |Z| + 1 > |X^{[f-]}| + 1 = |X|$$

is a contradiction to X being maximal consistent with $\mathcal{T}$. So $X^{[f-]}$ is maximal consistent with $\mathcal{T}_f$, hence induced by it. □

We can generalize this to flip sequences:

COROLLARY 2.3.3. *Let $\mathcal{T}$ be a 3-orientation of a 4-connected, triangular graph G. Let S be a flip sequence on $\mathcal{T}$ and let s_f denote how often a face f appears in S. Moreover let X and Y be the potentials induced by $\mathcal{T}$ resp. $\mathcal{T}_S$. Then we have $|Y| = |X| - |S|$ and it holds for all faces f*

$$y_f = x_f - s_f.$$

PROOF. By induction over $|S|$. The case for $|S| = 0$ is trivial and the induction step follows easily from Proposition 2.3.2. □

These results state that induced potentials change in a well-defined manner with flips of faces. In particular, let's denote the induced potential of a 3-orientation $\mathcal{T}$ with $\Pi(\mathcal{T})$. Then we have by the last corollary for any *flip* sequence S on a 3-orientation $\mathcal{T}$: $|\Pi(\mathcal{T}_S)| = |\Pi(\mathcal{T})| - |S|$.

We will now see that induced potentials describe exactly the rank of each face and the rank of a graph.

THEOREM 2.3.4. *Let G be a 4-connected, triangular graph G with a 3-orientation $\mathcal{T}$ and let X be the potential induced by $\mathcal{T}$. Then* $\text{RANK}(\mathcal{T}) = |X|$.

PROOF. By induction over $\text{RANK}(\mathcal{T})$. If the rank of $\mathcal{T}$ is zero, then there is no flippable face in $\mathcal{T}$. By Proposition 2.2.7 we have $x_f = 0$ for all faces f in the induced potential, thus $|X| = 0$.

So let the theorem be true for all 3-orientations with rank k, and $\mathcal{T}$ be a 3-orientation of rank $k+1$. Then there is a shortest maximal flip sequence of length $k+1$: let S be such a sequence and f_0 be the first face in S. Now consider $\mathcal{T}_{f_0}$ and its induced potential, denoted by X'. The rank of this 3-orientation$\mathcal{T}_{f_0}$ is k (clearly, S without the first f_0 is a maximal flip sequence of $\mathcal{T}_{f_0}$, and $\text{RANK}(\mathcal{T}_{f_0}) < k$ would contradict $\text{RANK}(\mathcal{T}) = k+1$), so by induction hypothesis $|X'| = k$. But from Proposition 2.3.2 it follows, that $|X| = |X'| + 1 = k + 1$. $\square$

THEOREM 2.3.5. *Let $\mathcal{T}$ be a 3-orientation of a 4-connected, triangular graph G and S be a maximal flip sequence on $\mathcal{T}$. Let s_f denote how often a face f appears in S. Then S has length* $\text{RANK}(\mathcal{T})$ *and $s_f = x_f = \text{RANK}^{\mathcal{T}}(f)$ for every face f.*

PROOF. Let X be the potential induced by $\mathcal{T}$. By definition of $\text{RANK}(\mathcal{T})$ we have $|S| \geq \text{RANK}(\mathcal{T})$ and by Theorem 2.3.4 this means $|S| \geq |X|$. The definition of the face rank and Lemma 2.1.3 imply

$$s_f \leq \text{RANK}^{\mathcal{T}}(f) \leq x_f \tag{2.3.1}$$

for each face f, hence $\sum_f s_f = |S| \leq |X|$ and therefore $|S| = |X| = \text{RANK}(\mathcal{T})$. This of course just means

$$\sum_f s_f = \sum_f x_f$$

which together with Equation 2.3.1 gives $s_f = \text{RANK}^{\mathcal{T}}(f) = x_f$.

So for each face the induced potential coincides with its rank. Let us rephrase: it does not matter which maximal flip sequence we choose, each face f will be flipped exactly x_f times. Some consequences from Theorem 2.3.5:

$\square$

- $\mathcal{L}_G$ clearly induces the minimal potential. The maximal value will be induced by $\mathcal{R}_G$ and any flip sequence S that leads from $\mathcal{R}_G$ to $\mathcal{L}_G$ is of length $\text{RANK}(\mathcal{R}_G)$,
- for every 3-orientation $\mathcal{T}$ it takes $\text{RANK}(\mathcal{T})$ flips to obtain $\mathcal{L}_G$ and

$$\text{RANK}(\mathcal{R}_G) - \text{RANK}(\mathcal{T})$$

flops to reach $\mathcal{R}_G$.

2.4. How to Find Inducing 3-Orientations

We will now turn to the question of how to reconstruct a 3-orientation from an induced potential X. From Lemma 2.2.2 we already know that there is exactly one 3-orientation inducing X. Therefore we will now intrinsically[1] characterize induced potentials so we can decide whether a potential is an induced one without actually knowing the inducing 3-orientation.

LEMMA 2.4.1. *Let G be a triangular, 4-connected graph. Then in any induced potential X of G and for any two adjacent faces f and g holds:*

$$x_f > x_g \quad \Rightarrow \quad f \textit{ is right of } g \textit{ in } \mathcal{L}_G \tag{2.4.1}$$

PROOF. Let $\mathcal{T}$ be the 3-orientation inducing X, and f and g be two adjacent faces with $x_f > x_g$ (which of course means $x_f = x_g + 1$). Now, X is in particular consistent with $\mathcal{T}$, which means that the higher potential value must be on the left side of the separating edge, so f is left of g. Hence, it is the face to be flipped first of both. Let now S be a maximal flip sequence on $\mathcal{T}$, so $\mathcal{T}_S = \mathcal{L}_G$. We will only consider flips of f and g in S. Because X is induced by $\mathcal{T}$, Theorem 2.3.5 gives that f will be flipped exactly x_f times, so g will be flipped $x_f - 1$ times. The interleaving of flips (Observation 1.7.4) states that the flips of f and g take turns, so the sequence of flips, neglecting all flips of faces $\notin \{f, g\}$, is

$$\underbrace{f, g, f, g, \ldots, g, f}_{x_f \text{ times } f,\ x_f - 1 \text{ times } g}$$

Thus so after its last flip, f is right of g, therefore it must also be right in $\mathcal{L}_G$, because no flip of g, which alone could change this situation, will occur thereafter. □

While this is a rather easy consequence of the fact that potential values describe numbers of flips, we will see now that the condition (Cond. 2.4.1) formulated in the above lemma already fully characterizes all maximal consistent potentials, independently of their inducing 3-orientation. That means we can decide whether a potential X is an induced one by only examining at $\mathcal{L}_G$ and X — and similar to consistency, we see that maximal consistency can also be checked *locally*, by checking the separating edges separately. In fact, we can even immediately construct the inducing 3-orientation from $\mathcal{L}_G$ and the heightcycles of X:

THEOREM 2.4.2. *Let G be 4-connected and triangular and let X be a potential on G. If for any two adjacent faces f and g $x_f > x_g$ implies that in $\mathcal{L}_G$ f is* right *of g (Condition 2.4.1 holds, i.e.), then X is an*

[1] nearly: we will still need $\mathcal{L}_G$.

induced potential. The inducing 3-orientation $\mathcal{T}_X$ can be constructed by setting the edge between f and g as follows:

- *if $x_f = x_g$, then orient e as in $\mathcal{L}_G$,*
- *if $x_f > x_g$, then orient e such that f is left of g (i.e. reverse it with respect to $\mathcal{L}_G$.)*

PROOF. To see that the obtained orientation is indeed a 3-orientation, note that the construction of $\mathcal{T}_X$ tells us to reverse all edges in $\mathcal{L}_G$ that are separating in X. We claim that the separating edges of X together form a set of directed cycles in $\mathcal{L}_G$: let R be a k-region in X for any $k > 0$. X satisfies Condition (2.4.1) which implies that the outer boundary of R is a cw-cycle in $\mathcal{L}_G$, while possible inner boundaries are ccw-cycles, using the same reasoning as in Lemma 2.2.4. So reversing separating edges means reversing the direction of oriented cycles in $\mathcal{L}_G$ (These cycles are edge-disjoint because the heightcycles in X are.) Hence, the result will again be a 3-orientation (1.3.1).

So we need to show that $\mathcal{T}_X$ induces X. We use induction on $|X|$: if $|X| = 0$, then there are no separating edges, thus $\mathcal{T}_X = \mathcal{L}_G$ and Proposition 2.2.7 states that $\mathcal{L}_G$ induces the zero potential.

So let $|X| = n$ and the statement be true for all potentials with a value less than n. Morever, let $k > 0$ be the largest value appearing in X and f be a face with value k. Then f is contained in a k-region R and since k is the maximal value, all faces in R must have value k and R is simple.

We will now construct a potential X' by lowering all faces in R by one. This will enable us to apply the induction hypothesis. Comparing $\mathcal{T}_X$ with $\mathcal{T}_{X'}$ then leads to the desired result.

So X' is defined in the obvious way:

$$x'_f := \begin{cases} x_f - 1 = k - 1 & \text{if } f \in R \\ x_f & \text{otherwise} \end{cases}$$

Checking the edges on the boundary of R shows that X' indeed is a potential, and, moreover, it satifies Condition (2.4.1): $x'_f > x'_g$ implies $x_f > x_g$ from which in turn follows that f is right of g in $\mathcal{L}_G$. By the induction hypothesis X' is an induced potential, and the inducing 3-orientation, denoted by $\mathcal{T}_{X'}$, is constructed as stated above. Now let $\mathcal{T}_X$ be constructed accordingly. Clearly, X is consistent with $\mathcal{T}_X$.

Consider the boundary of R. By Lemma 2.2.4 it is a ccw-cycle in $\mathcal{T}_X$ and we have already seen that it is a cw-cycle in $\mathcal{L}_G$ because R is a k-region. However, this implies that it is a cw-cycle in $\mathcal{T}_{X'}$: by construction of X', there is no potential difference along R's boundary. So all its edges must be oriented as in $\mathcal{L}_G$ and thus form a cw-cycle, too. Furthermore, it is easy to see that these are the only edges where

$\mathcal{T}_X$ and $\mathcal{T}_{X'}$ differ: an edge not bounding R will be separating either in both X and X' or in none of the two. If it *is* separating, the greater value will in both potentials be on the same side. Hence, $\mathcal{T}_X$ and $\mathcal{T}_{X'}$ are identical on the remaining edges.

This means that $\mathcal{T}_X$ and $\mathcal{T}_{X'}$ differ only in the direction of one directed cycle. Let c denote the number of faces in R: then we know from Proposition 1.7.3 that $\mathcal{T}_{X'}$ can be obtained from $\mathcal{T}_X$ by c flips. Corollary 2.3.3 states that in this case the values of the induced potentials differ exactly by c. Since $\mathcal{T}_{X'}$ is induced by X' this gives a value of $|X'| + c$ for the induced potential of $\mathcal{T}_X$. But obviously $|X| = |X'| + c$ so X has already maximal value, so it must be the induced potential (see Corollary 2.1.5), which completes our proof. $\square$

Note that another shorter way to describe the above construction is this: to get $\mathcal{T}_X$, take the heightcycles of X and reverse the respective edges in $\mathcal{L}_G$.

This was an important step: we are now able not only to get an induced potential from a 3-orientation, but we can also decide whether a given potential is induced and in this case construct the inducing 3-orientation. More formally:

THEOREM 2.4.3. *There is a bijective mapping between the set of all 3-orientations and the set of all induced potentials, defined by*

$$\Pi : \mathcal{T} \mapsto X,\ X := \textit{the potential induced by } \mathcal{T}$$

and the inverse Π^{-1} *is given by* $X \mapsto \mathcal{T}_X$.

PROOF. Π is surjective by definition and injective by Lemma 2.2.2. By construction $\mathcal{T}_X$ induces X, which in terms of Π can also be stated as $\Pi(\mathcal{T}_X) = X$. But then $X \mapsto \mathcal{T}_X$ must be the inverse of Π. $\square$

CHAPTER 3

The Lattice of 3-Orientations

"When all questions of space, time, and matter
and the nature of being have been resolved, only
one question remains – 'Where shall we have dinner?'"

D. Adams
The Restaurant at the End of the Universe

In this chapter we will try to get a deeper insight into the structure of the set of all 3-orientations of a 4-connected graph — and we will see that it is richely structured, namely that all 3-orientations not only form a poset, but a distributive lattice. In particular, there is a unique maximal element and unique minimal element, called $\hat{1}$ resp. $\hat{0}$, which are exactly the cw-cycle-free resp. the ccw-cycle-free 3-orientations $\mathcal{R}_G$ and $\mathcal{L}_G$.

3.1. Ordering Orientations

Definition 3.1.1. Let G be a triangular graph. Define

- $\mathcal{O}^3(G) := \{\mathcal{T} : \mathcal{T}$ is a 3-orientation of $G\}$
- for two 3-orientations $\mathcal{T}$ and $\mathcal{S}$ is $\mathcal{S} \preceq \mathcal{T} :\Leftrightarrow \mathcal{S}$ is obtainable from $\mathcal{T}$ by flipping some faces, i.e. there exists a *flip sequence* S, such that $\mathcal{T}_S = \mathcal{S}$.

Proposition 3.1.2. *Let G be a 4-connected, triangular graph. Then $\preceq$ is a partial order on $\mathcal{O}^3(G)$.*

Proof. This is straightforward - in the following let $\mathcal{R}, \mathcal{S}, \mathcal{T}$ be 3-orientations of the same 4-connected, triangular graph G :

Reflexivity: if S is an empty sequence, then $\mathcal{T}_S = \mathcal{T}$, so $\mathcal{T} \preceq \mathcal{T}$

Transitivity: Let $\mathcal{R} \preceq \mathcal{S}$ and $\mathcal{S} \preceq \mathcal{T}$. Then there exist flipsequences S and S' with $\mathcal{R} = \mathcal{S}_S$ and $\mathcal{S} = \mathcal{T}_{S'}$. If U denotes the concatentation of S' and S, then clearly $\mathcal{R} = \mathcal{T}_U$.

Antisymmetry: If $\mathcal{S} \preceq \mathcal{T}$ and $\mathcal{T} \preceq \mathcal{S}$, then there are flipsequences S and S' with $\mathcal{S} = \mathcal{T}_S$ and $\mathcal{T} = \mathcal{S}_{S'}$. Then $U = SS'$ yields a flip sequence with $\mathcal{T}_U = \mathcal{T}$, so U must be empty, because otherwise we would by repeated concatenation be able to produce an arbitrarily long flip sequence on $\mathcal{T}$, which contradicts Corollary 1.7.6. Thus S and S' are empty, too, and $\mathcal{T} = \mathcal{S}$ follows.

$\square$

COROLLARY 3.1.3. *Given a 4-connected, triangular graph G, the poset $(\mathcal{O}^3(G), \preceq)$ is graded with the rank function $p : \mathcal{O}^3(G) \ni \mathcal{T} \mapsto \mathrm{RANK}(\mathcal{T})$. Moreover, it has $\hat{1}$ and $\hat{0}$, given by the 3-orientations $\mathcal{R}_G$ resp. $\mathcal{L}_G$.*

PROOF. The latter part is clear. For p to be a rank function it needs to meet the following criteria: $p(\mathcal{T}) = 0$ if $\mathcal{T}$ is minimal in $\mathcal{O}^3(G)$ and $p(\mathcal{S}) = p(\mathcal{T}) + 1$ if $\mathcal{S}$ covers[1] $\mathcal{T}$ in $\mathcal{O}^3(G)$. However, this immediately follows from the results of the last chapter and the definition of p. $\square$

These conditions are only necessary for $\mathcal{O}^3(G)$ to be a distributive lattice but nevertheless interesting. To actually prove that $\mathcal{O}^3(G)$ *is* a distributive lattice, we will first show that the set of induced potentials forms a distributive lattice on its own and then use the bijection Π we gave in Theorem 2.4.3.

DEFINITION 3.1.4. Let G be a 4-connected and triangular graph. Define

- $\mathcal{P}(G) := \{X : X \text{ is an induced potential on } G\}$
- for two induced potentials X, Y is

$$X \sqsubseteq Y \quad :\Leftrightarrow \quad x_f \leq y_f \text{ for all faces } f$$

Remember that when we first introduced the maximum (MAX) and the minimum (MIN) of two potentials in section 2.1, we would only combine potentials consistent with the *same* 3-orientation $\mathcal{T}$ to get a new potential, which again would be consistent with $\mathcal{T}$. What we claim now is that combining any two induced potentials — each one consistent with different 3-orientations — again yields an induced potential:

LEMMA 3.1.5. *Let G be a 4-connected, triangular graph. Then for $X, Y \in \mathcal{P}(G)$ is $\mathrm{MAX}(X, Y) \in \mathcal{P}(G)$ and $\mathrm{MIN}(X, Y) \in \mathcal{P}(G)$. $\mathrm{MAX}(X, Y)$ is the least upper bound of X and Y, while $\mathrm{MIN}(X, Y)$ is the greatest lower bound of X and Y.*

PROOF. We will only give the proof for $\mathrm{MAX}(X, Y)$, it works the same way for $\mathrm{MIN}(X, Y)$. For the first part we will use the characterization for induced potentials introduced in Lemma 2.4.1. Therefore we only need to look at the separating edges in $\mathrm{MAX}(X, Y)$.

Let $Z := \mathrm{MAX}(X, Y)$, i.e. for all faces f we have $z_f := \max(x_f, y_f)$. Let now f and g be adjacent faces with $z_f > z_g$.We have to show that f

[1]For a poset P an element a is said to be *covering* b if $a > b$ and there exists no c s.th. $a > c > b$.

is right of g in $\mathcal{L}_G$. Obviously not both $x_f \leq x_g$ and $y_f \leq y_g$ can hold, because this would imply $z_f \leq z_g$. So w.l.o.g. $x_f > x_g$. Since X itself is induced, this implies that f is right of g in $\mathcal{L}_G$ (Condition 2.4.1). Hence, $\textsc{max}(X, Y)$ is an induced potential and it follows immediately from the face-wise definition of $\textsc{max}$ that it is indeed the least upper bound. □

We are now well prepared to approach our main result:

PROPOSITION 3.1.6. *Given a 4-connected, triangular graph G, $\mathcal{P}(G)$ is a distributive lattice, where the join $(X \vee Y)$ is given by* $\textsc{max}(X, Y)$ *and* $\textsc{min}(X, Y)$ *is the meet $(X \wedge Y)$.*

PROOF. From Lemma 3.1.5 we know that join and meet as above are indeed well-defined least upper and greatest lower bound. Due to the definition of $\textsc{max}$ and $\textsc{min}$ on potentials via max resp. min of face potential values we get distributivity for free since min and max are face-wise distributive. □

As a last step we will show that the bijection $\Pi : \mathcal{O}^3(G) \to \mathcal{P}(G)$ is an order isomorphism. It is then an immediate consequence that with $\mathcal{P}(G)$ also $\mathcal{O}^3(G)$ is a distributive lattice.

LEMMA 3.1.7. *Given a 4-connected, triangular graph G and the corresponding posets $(\mathcal{O}^3(G), \preceq)$ and $(\mathcal{P}(G), \sqsubseteq)$. Then Π is order-preserving, i.e.*

$$\mathcal{S} \preceq \mathcal{T} \Leftrightarrow \Pi(\mathcal{S}) \sqsubseteq \Pi(\mathcal{T})$$

PROOF. "$\Rightarrow$" — You can obtain $\mathcal{S}$ from $\mathcal{T}$ by face flips, which will decrease the potential value of the flipped faces. So $\Pi(\mathcal{S}) \sqsubseteq \Pi(\mathcal{T})$.

"$\Leftarrow$" — Let $X := \Pi(\mathcal{S})$ and $Y := \Pi(\mathcal{T})$ with $X \sqsubseteq Y$. The proof is by induction on $|Y| - |X|$. If $|X| = |Y|$, there is nothing to prove. If not, let R be a maximal region of faces f with $x_f < y_f$. We claim that R contains a flippable face g, and flipping this face will allow us to apply the induction hypothesis. Because then set $Y' := \Pi(\mathcal{T}_g) = Y^{[g-]}$ (see Proposition 2.3.2.) Clearly $|Y'| = |Y| - 1$, and since Y and Y' differ only in the value of g, $X \sqsubseteq Y'$ also holds because $x_g < y_g$ implies $x_g \leq y'_g$. Then, using the induction hypothesis, we have

$$\mathcal{T}_X \preceq \mathcal{T}_{Y'} = \mathcal{T}_g \preceq \mathcal{T}.$$

It remains to show that R does indeed contain a flippable face. We will only show, that R is a *ccw-bounded* region, that is, whenever two faces $f \in R$ and $g \notin R$ are adjacent, then f is left of g[2], and then

[2]This in particular implies that the *outer* boundary of R is a ccw-cycle, and we know from Proposition 1.7.3 that such a cycle always contains a flippable face. But unfortunately R could contain holes and we must be sure that the flippable face really is *in* R.

claim that any such region contains a flippable face. The argument why this is the case is similar to the one we used to show that we find a flippable face in any ccw-cycle[3] (see Prop. 1.7.3), but we have to take into account possible holes in R, which makes things a bit more complicated. Therefore you can find the complete proof in Appendix .1.2.

To show that R is ccw-bounded, let f and g be adjacent faces, f inside and g outside R. By construction $x_f < y_f$ while $x_g = y_g$.

CASE 1. $y_f > y_g$ immediately implies that f is right of g in $\mathcal{T}$.

CASE 2. $y_f < y_g$: this case is not possible, because $x_f < y_f < y_g = x_g$ would imply $x_g - x_f > 1$.

CASE 3. Finally, if $y_f = y_g$ holds, we know from Theorem 2.4.2 that in $\mathcal{T}$, which induces Y, the orientation of the edge between f and g is the same as in $\mathcal{L}_G$. Now Lemma 2.4.1 states that due to $x_f < y_f = y_g = x_g$ in $\mathcal{L}_G$ g is right of f, so f is left of g. $\square$

COROLLARY 3.1.8. *Given a 4-connected triangular graph G, the set of 3-orientations $\mathcal{O}^3(G)$ together with the ordering by flips $\preceq$, is a distributive lattice.*

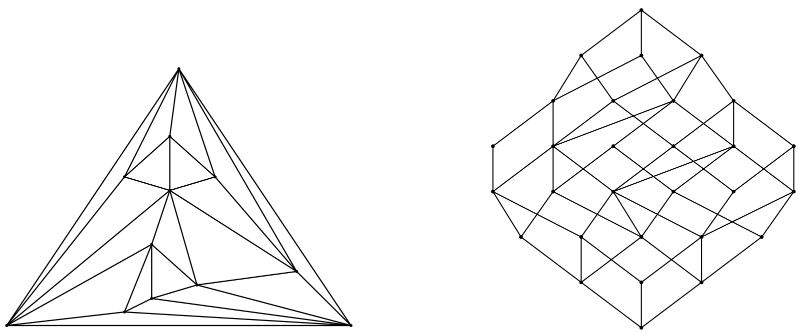


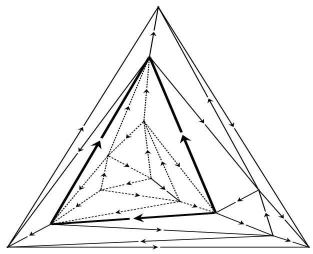
FIGURE 3.1.1. A 4-connected graph and the corresponding distributive lattice

3.2. Closing The Cycle

In 4-connected graphs the structure of the set of 3-orientations was explored rather easily. To obtain results from this for the case of general triangular graphs we will exploit some nice inductive structure which connects 4-connected triangular graph to those not 4-connected: the latter can always be split into two triangular graphs containing less separating triangles while the former are the base case about which we

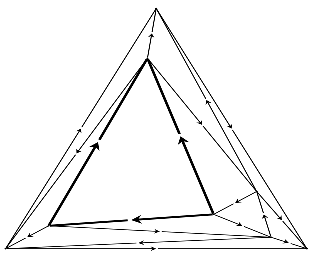
[3]In fact, the interior of a ccw-cycle *is* a ccw-bounded region.

already obtained results. (Don't get confused by *separating* edges and *separating* triangles: they have nothing in common.)

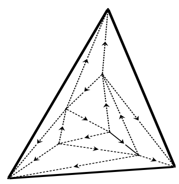


(a) G with $\mathcal{T}$ (edges of C bold)

split to



(b) $G \setminus C$ with $\mathcal{T} \setminus C$



(c) $G|_C$ with $\mathcal{T}|_C$

FIGURE 3.2.1. Splitting a graph with 3-orientation

DEFINITION 3.2.1. Let G be a triangular graph and s be a separating triangle in G. Then define

$$\begin{aligned} G \setminus s &:= G \setminus \{v : v \text{ is a node from the interior of } s\} \\ G|_s &:= G \setminus \{v : v \text{ is a node from the exterior of } s\} \end{aligned}$$

(See Figure 3.2.1.)

Obviously, both graphs themselves are triangular, and both contain s. In this example the edges of $G \setminus s$ and $G|_s$ are oriented as in G which in turn is 3-oriented. A closer examination of the picture will show that both $G \setminus s$ and $G|_s$ are themselves 3-oriented. We will now see that this is no accident:

PROPOSITION 3.2.2 (Orthogonality). *Let G be a triangular graph with a 3-orientation $\mathcal{T}$ and let s be a separating triangle in G. Let the interior edges of $G \setminus s$ and $G|_s$ be oriented as they are in $\mathcal{T}$. Then both graphs are 3-oriented and we will denote the respective 3-orientations by $\mathcal{T} \setminus s$ and $\mathcal{T}|_s$.*

The converse also holds: given orientations $\mathcal{T}$ on $G|_s$ and $\mathcal{S}$ on $G \setminus s$, we obtain a 3-orientation for G by orienting each interior edge as it is in $\mathcal{T}$ resp. $\mathcal{S}$. We denote this 3-orientation by $\mathcal{T} \oplus \mathcal{S}$.

PROOF. Every interior node of $G|_s$ is an interior node of G and all its adajcent edges are also interior edges of $G|_s$. Clearly $d^+(v) = 3$ in $\mathcal{T}$ implies $d^+(v) = 3$ in the specified orientation of $G|_s$, so $G|_s$ is 3-oriented itself.

On the other hand this implies that there is no interior edge of s leaving a node on s (see Prop. 1.1.1). Therefore, by removing from G all inner nodes and edges of s the nodes on s will not loose an outgoing edge. Hence, their outdegree will not change and $G \setminus s$ is 3-oriented, too.

For the converse, note that any interior edge of G will be an interior edge either in $G|_s$ or $G \setminus s$ but not in both, so its orientation is well-defined. The same reasoning as above shows that then indeed every interior node in G has outdegree 3. $\square$

COROLLARY 3.2.3. *For every triangular graph G with a separating triangle s there exists a bijection between $\mathcal{O}^3(G)$ and $\mathcal{O}^3(G \setminus s) \times \mathcal{O}^3(G|_s)$.*

PROOF. The mapping

$$p: \begin{array}{ccc} \mathcal{O}^3(G) & \to & \mathcal{O}^3(G \setminus s) \times \mathcal{O}^3(G|_s) \\ \mathcal{T} & \mapsto & (\mathcal{T}|_s, \mathcal{T} \setminus s) \end{array} \tag{3.2.1}$$

is bijective by Theorem 3.2.2 with inverse function $(\mathcal{T}, \mathcal{T}') \mapsto \mathcal{T} \oplus \mathcal{T}'$. $\square$

We have extensively seen that the smallest possible modification we can apply to a 3-orientation is the reversal of triangles, which in 4-connected nicely corresponded to a face. However, now we have to drop the notion of *flipping faces*, and instead have to *flip triangles,* but this shift is going to be a very natural one. As we will see, the basic concepts will be transferable to general triangular graphs.

DEFINITION 3.2.4. For a triangular graph G define

$$\Delta(G) := \{t : t \text{ is a triangle of } G\}.$$

Let $\mathcal{T}$ be a 3-orientation of G. Analogous as we did for the faces, we call a triangle t *flippable* if in $\mathcal{T}$ it is a ccw-triangle, and *floppable* if it is a cw-triangle. *(Triangle) flip* resp. *flop sequences* are defined accordingly (see Def. 1.7.2.)

We will see now that, given a separating triangle s, we can not only split graphs and 3-orientations but also flip sequences, namely into these triangle flips that modify the 3-orientation of $G|_s$ and those that modify the 3-orientation of $G \setminus s$. Note that for a flip of a triangle t to modify the 3-orientations of both graphs, t would have to consist of interior

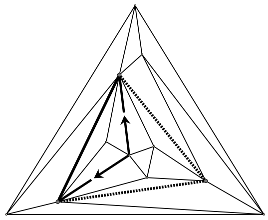


FIGURE 3.2.2.

edges from both graphs. Then, however, this triangle t basically must look like the one in Fig. 3.2.2 (the outer bold edges mark s while the non-dotted bold edges mark t). Theorem 3.2.2 imposes the directions of the edges shown in the picture because the edges are adjacent to exterior nodes of $G|_s$. Thus t can never be flippable. Therefore we can partition a flip sequence on a 3-orientation $\mathcal{T}$ in those flips modifying $\mathcal{T}|_s$ and those modifying $\mathcal{T} \setminus s$.

DEFINITION 3.2.5. Let G be a triangular graph with a separating triangle s and let S be a flip sequence on some 3-orientation of G. Then let

$$\begin{aligned} S|_s &:= \{t \in S : t \text{ is an interior triangle in } G|_s\} \\ S \setminus s &:= \{t \in S : t \text{ is an interior triangle in } G \setminus s\} \end{aligned}$$

Note that here, *interior triangle* here means all triangles except the one formed by the boundary of the exterior face.

The fact that 3-orientations of $G|_s$ and $G \setminus s$ are independent of each other implies that for a given flip sequence S on a 3-orientation $\mathcal{T}$ we have

$$\mathcal{T}_S = (\mathcal{T}|_s)_{(S|_s)} \oplus (\mathcal{T} \setminus s)_{(S \setminus s)}$$

that is, to obtain $\mathcal{T}_S$ we could also partition S into $(S|_s, S \setminus s)$ and flip triangles separately in $\mathcal{T}|_s$ and $\mathcal{T} \setminus s$. Accordingly, if we have flip sequences S and S' for $\mathcal{T}|_s$ resp. $\mathcal{T} \setminus s$, then SS' will be a flip sequence on $\mathcal{T}$.

Face flips impose a partial order on the 3-orientations of 4-connected graphs. As we will see now, triangle flips do the same for general triangular graphs. Rest assured that this will not render our preceeding results worthless: we will need them as the base case for induction, like for example in the proof of the next proposition.

PROPOSITION 3.2.6. *Let G be a triangular graph. Define a relation $\leq$ as follows: for two 3-orientations $\mathcal{T}$ and $\mathcal{S}$ let $\mathcal{S} \leq \mathcal{T} :\Leftrightarrow \mathcal{S}$ is obtainable from $\mathcal{T}$ by flipping some triangles, i.e. there exists a* (triangle) flip sequence *S, such that $\mathcal{T}_S = \mathcal{S}$. Then $\leq$ is a partial order on $\mathcal{O}^3(G)$ (comp. Def. 3.1.1.)*

PROOF. Clearly, $\leq$ is reflexive and transitive. We will prove its antisymmetry by induction over the number of separating triangles in G. If G is 4-connected, then $\Delta(G)$ describes exactly the set of faces of G and $\leq$ is equivalent to $\preceq$, so there is everything proved, see Prop. 3.1.2. If G is not 4-connected, then it contains a separating triangle s. Let now $\mathcal{S} \leq \mathcal{T}$ and $\mathcal{T} \leq \mathcal{S}$. By definition there exist flip sequences S and S' such that

$$\mathcal{S} = \mathcal{T}_S \text{ and } \mathcal{T} = \mathcal{S}_{S'}$$

Then we know

$$\mathcal{S}|_s = (\mathcal{T}|_s)_{S|_s} \text{ and } \mathcal{T}|_s = (\mathcal{S}|_s)_{S'|_s}$$

thus by induction hypothesis $\mathcal{S}|_s = \mathcal{T}|_s$. Analogously we get $\mathcal{S} \setminus s = \mathcal{T} \setminus s$, hence $\mathcal{S} = \mathcal{T}$. □

We will now see that this order respects orthogonality, i.e. it is isomorphic to the order obtained by splitting the graph G along some sepearating triangle s and taking the order induced[4] on $\mathcal{O}^3(G)$ by $(\mathcal{O}^3(G|_s), \leq) \times (\mathcal{O}^3(G \setminus s), \leq)$. In particular, this implies that for any two separating triangles s and s' the posets

$$(\mathcal{O}^3(G|_s), \leq) \times (\mathcal{O}^3(G \setminus s), \leq)$$

resp.

$$(\mathcal{O}^3(G|_{s'}), \leq) \times (\mathcal{O}^3(G \setminus s'), \leq)$$

are isomorphic.

LEMMA 3.2.7. *Let G be a triangular graph. If s is a separating triangle in G, then*

$$(\mathcal{O}^3(G), \leq) \simeq (\mathcal{O}^3(G|_s), \leq) \times (\mathcal{O}^3(G \setminus s), \leq)$$

PROOF. We will show that the bijection defined in (3.2.1) is an order isomorphism. So let s be a separating triangle of G, and $\mathcal{T}$, $\mathcal{S}$ be two 3-orientations of G with $\mathcal{T} \leq \mathcal{S}$. Then there exists a flip sequence S on G, such that $\mathcal{T} = \mathcal{S}_S$. This implies

$$\begin{array}{cccc} \Rightarrow & \mathcal{T}|_s = (\mathcal{S}|_s)_{S|_s} & \wedge & \mathcal{T} \setminus s = (\mathcal{S} \setminus s)_{S \setminus s} \\ \Rightarrow & \mathcal{T}|_s \leq \mathcal{S}|_s & \wedge & \mathcal{T} \setminus s \leq \mathcal{S} \setminus s \\ \Rightarrow & (\mathcal{T}|_s, \mathcal{T} \setminus s) & \leq & (\mathcal{S}|_s, \mathcal{S} \setminus s). \end{array}$$

Let now $\mathcal{T}$ and $\mathcal{S}$ be 3-orientations of $G|_s$ and $\mathcal{T}$' and $\mathcal{S}$' be 3-orientations of $G \setminus s$, such that $\mathcal{T} \leq \mathcal{S}$ and $\mathcal{T}' \leq \mathcal{S}'$. Then there exist flip sequences S and S' on $G|_s$ resp. $G \setminus s$ such that$\mathcal{T} = \mathcal{S}_S$ and $\mathcal{T}' = \mathcal{S}'_{S'}$. Then we have

$$\begin{array}{rcl} \mathcal{T} \oplus \mathcal{T}' & = & (\mathcal{S}_S \oplus \mathcal{S}'_{S'}) \\ & = & (\mathcal{S} \oplus \mathcal{S}')_{SS'} \end{array},$$

[4] The cartesian product $A \times B$ of two posets A,B is again a poset, if you set $(a, b) \leq (a', b') :\Leftrightarrow a \leq a' \wedge b \leq b'$ for $a, a' \in A$ and $b, b' \in B$. See [**9**].

and this implies

$$\mathcal{T} \oplus \mathcal{T}' \leq (\mathcal{S} \oplus \mathcal{S}').$$

□

The main theorem of this chapter is then obtained by an easy induction, again over the number of separating triangles:

THEOREM 3.2.8. *Let G be a triangular graph. Then $(\mathcal{O}^3(G), \leq)$ is a distributive lattice.*

PROOF. If G is 4-connected then everything is already proved by Corollary 3.1.8. Otherwise, let s be a separating cycle of G. By induction hypothesis $(\mathcal{O}^3(G|_s), \leq)$ and $(\mathcal{O}^3(G \setminus s), \leq)$ are distributive lattices. Then lattice theory gives that $(\mathcal{O}^3(G|_s, \leq) \times (\mathcal{O}^3(G \setminus s), \leq))$ also is a distributive lattice, which by Lemma 3.2.7 is isomorphic to $(\mathcal{O}^3(G \setminus s), \leq)$. So $(\mathcal{O}^3(G \setminus s), \leq)$ is a distributive lattice itself. □

3.3. Where To Go From Here

There is still a lot of open questions. Although we proved that the 3-orientations of triangular graphs form a distributive lattice we did not say what join and meet in these lattices actually look like. Even in the — apparently simpler — case of 4-connected, triangular graphs join and meet were only declared in terms of the isomorphic lattice of induced potentials. So in some sense our proof is only a proof of existence and some work remains to be done. It might help to generalize the concept of *potentials* to general triangular graphs. However, by doing this we might loose the intuition of potential values describing the number of flips possible on a face. Two possible ways to do this are: define potentials over $\Delta(G)$, the set of all triangles of a graph G. This set seems to be the natural generialization of the the set of faces of 4-connected graphs, as we saw in the last section. Another possibility which seems promising is to stick to potentials over the faces as we have seen them, and try to find a different intuition. We conjecture that there is a strong connection between maximal potential values and the number of edge-disjoint directed cycles in which a face is contained.

Another question to ask is: given a distributive lattice L, is there a triangular graph G, such that $\mathcal{O}^3(G) \simeq L$. We have seen that a trivial lattice, consisting of one element only, describes the structure of the 3-orientations of every stacked graph. On the other hand it is not very hard to show that by nesting n separating triangles the way shown in Fig. 3.3.1, we get a graph G such that $\mathcal{O}^3(G) \simeq B_n$, where B_n is the Boolean Algebra formed by the subsets of a set with n elements, ordered by inclusion. This gives also some hints on the number of possible 3-orientations which in the latter case is $O(2^n)$. A similar question is whether sublattices of $\mathcal{O}^3(G)$ have an interpretation in terms of G, its faces and flips.

If you take a look at the application described in Chapter 5 you might note that the embedded 3-orientations generated with the "random" setting do not always look very random — there tend to be more nodes "on the bottom" of the graph than there are nodes near the top node. So even if the algorithm described in the next chapter is capable of generating all possible 3-orientations resp. Schnyder labellings it seems to be far from generating them in a uniformly distributed way, and it is not clear how that could be done efficiently.

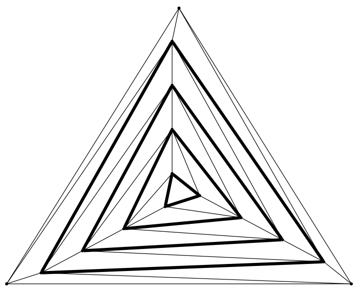


FIGURE 3.3.1. Nested separating triangles

Another interesting issue is the *generating poset*: given a poset P, we can order its set of ideals[5] by inclusion. What we get this way is a distributive lattice. The fundamental theorem for finite distributive lattices states that *every* such lattice is *generated* this way [**9**]. It it is possible to gain the generating poset from the lattice. First experiments (see Section 5.1.5) indicated that the generating poset of the distributive lattice of all 3-orientations seems to have interesting interpretations in terms of face flips, and it should be able to describe the whole structure of face flips in a very compact manner. In general, the connection to order theory seems to open a wide range of possibilities.

When we are going to create and display distributive lattices of 3-orientations, we hit another problem: since we are actually performing a depth-first search on the *graph of all 3-orientations* we needed to quickly decide whether we have seen a 3-orientation $\mathcal{T}$ before. So it is necessary to find an encoding of a given 3-orientation. Although the induced potential obviously is one, it is not very compact. Given a fixed orientation of the edges, we can get away with one bit per edge to store whether the edge is directed identical to the given orientation. But this approach does not take into account any properties of 3-orientations and would work for arbitrary orientations of graphs. So the question is: can we do better? Experiments showed that it is sufficient to store for any face whether it is flippable, floppable or none of both, though we did not prove it. Unfortunately, although this approach will no longer work for general orientations, the encoding delivered by this idea is not

[5]An *ideal* is a a set $I \subseteq P$ such set for all $x \in P$ holds: if $x \leq y$, $y \in I$, then $x \in I$.

even as compact as the one mentioned before. However, there might be some connection to oriented matroids here. These can be encoded in a similar fashion by vectors of $\{+, -, 0\}$, and as in our case, there is structure in the set of valid encodings. If it was possible to characterize the set of all valid encodings completely, this might lead to a way to construct Schnyder labellings more "randomly".

CHAPTER 4

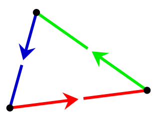
On Schnyder Labellings

"And now for something completely different . . . "
Monty Python

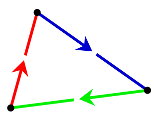
4.1. Face Flips & Schnyder Labellings

We have seen that we can obtain new 3-orientations from old ones by reversing directed cycles, which in turn can be mimicked by flipping the faces contained in a directed cycle (Prop. 1.7.3). But what if we do not only want to consider 3-orientations but Schnyder labellings, i.e. we have additional color information on the edges as defined in Section 1.2.1, which we want to keep up to date with each flip?

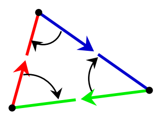
A look at the constraints that are put upon the colors of the incoming and outgoing edges of a node in a Schnyder labelling (see Fig. 1.2.2(a)) reveals that the edges of flippable or floppable faces can not have arbitrary colors. In fact, they will — up to *cyclic* rotation of the colors — always appear as displayed in Fig. 4.1.1(a) and 4.1.1(b).



(a) flippable



(b) floppable



(c) a face flop

FIGURE 4.1.1. Flippable/floppable faces in Schnyder labellings

Fortunately, we need to change the colors of the reversed edges only to maintain a Schnyder labelling. Let RED, GREEN, BLUE be a cyclic order of the colors, i.e. GREEN comes after RED, BLUE after GREEN, and RED after BLUE. Figure 4.1.1 already tells us how to do a flip on a face f in a Schnyder labelling: for each edge, reverse its direction (as usual) and give it the succeeding color with respect to the cyclic order above. A more visual description of how to do a flop is given in Fig 4.1.1(c).

It is not hard to verify that for each node on the boundary of f the in- and outgoing edges are still correctly colored to satisfy Definition 1.2.1 (without proof). Hence, the result is again a Schnyder labelling. It is clear that using this modified flip we can reverse any directed cycle and maintain a Schnyder labelling. However, we can describe the change in the colors caused by such a reversal in a shorter way:

LEMMA 4.1.1. *Let G be a triangular graph with a Schnyder labelling $\mathcal{S}$, that is, a 3-orientation plus a color for each interior edge of G. Let C be a ccw-cycle (cw-cycle) in $\mathcal{S}$. Then we obtain a new Schnyder labelling of G by*

1. *reversing the direction of the edges of C,*
2. *setting for each edge* on *C the new color to be the color succeeding (preceding) its original color,*
3. *setting for each edge* inside *C the new color to be the color preceding (succeeding) its original color,*
4. *leaving the color of any other edge unchanged.*

PROOF. We know from Proposition 1.7.3 that we can simulate the reversal of C by flipping each face inside C once. This means that every edge inside C participates in exactly two flips: those of its two adjacent faces. Each time it will be recolored with its respective succeeding color. Due to the cyclic ordering of the colors, after two recolorings it will have the color that *precedes* its initial color. The same argument applies to the edges on C. Such an edge participates only in one flip, so the resulting color will be the one succeeding its initial color (See Figure 4.1.2). ☐

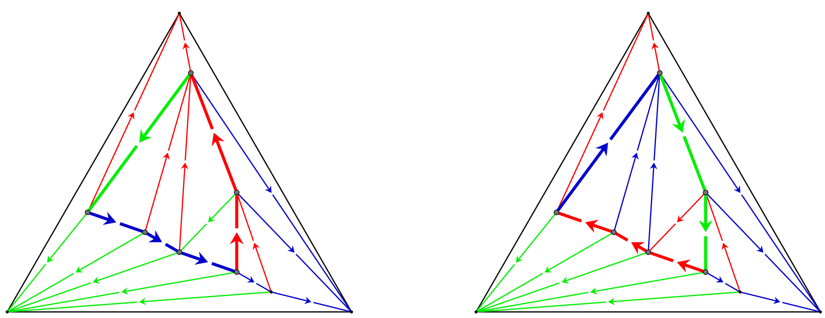
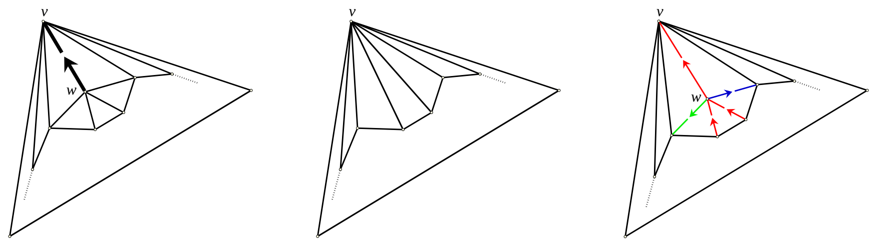


FIGURE 4.1.2. Reversing cycles in Schnyder labellings

4.2. Constructing Schnyder Labellings

Schnyder presented in [**8**] an algorithm to construct Schnyder labellings in linear time, mainly to create embeddings of triangular graphs. The algorithm we present here is only a variant of Schnyder's algorithm, but we think it is easier to implement since it is not based on edge contractions. These are — depending on the data structure you are using — a bit tedious to handle. Moreover, our algorithm shows how to construct the 3-orientations without clockwise resp. counterclockwise cycles. Since this algorithm is similar to Schnyder's it would also be possible to let the original algorithm to do that. Since we can not expect better running times for constructing pure 3-orientations than linear time, this seems also appropriate to construct 3-orientations, even if you are not interested in the coloring of the edges. We will first outline how the original algorithm works and then present our variant.

The basic idea of Schnyder's algorithm is to contract edges incident to exterior nodes until the graph is a single triangle and then expanding the edges again in reverse order. During the expansion, colors and directions are assigned to the edges as they reappear. An edge e, incident to an exterior node v and an interior node w, is *contractible* if v has exactly two common neighbors with w. (See Fig. 4.2.1(a)). When e will be re-expanded, w reappears and its edges will be colored as shown in Fig. 4.2.1(c).



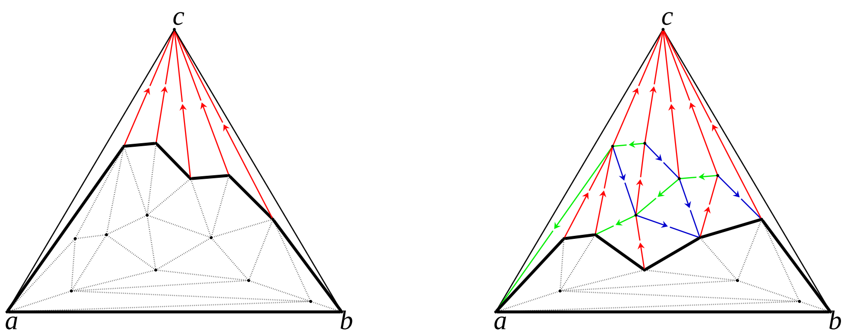
(a) before contraction (b) after contraction (c) after re-expansion

FIGURE 4.2.1. Edge contraction

It is part of Schnyder's work in [**8**] to show that in every triangular graph such a contractible edge always exists (in fact, there exists one at *each* exterior node). Since contraction always leaves again a triangular graph, it is clear that we indeed are able to repeat contracting edges until only the outer triangle of the graph is left. From Fig. 4.2.1(c) we can assure ourselves that this algorithm yields a 3-orientation: during each expansion step exactly one node reappears and at that point it will get three outgoing edges. Moreover, it is clear that at this point its incident edges are colored correctly (cmp. Def. 1.2.1), and it is not very hard to see that the edges appearing later will fit into the coloring of the edges that appeared before, for details see [**8**].

Schnyder showed that you can construct every possible Schnyder labelling this way. Even if you choose an arbitrary exterior node and contract edges toward this node only, you will still be able to get all possible Schnyder labellings.

This is a result that we will exploit in our variant of the construction algorithm. We will, as Schnyder does, always assign colors and orientations simultanously, so when we speak of "directing an edge", this includes coloring it and vice versa. The basic idea is to avoid contractions and instead keep up a cycle C, so that for nodes outside this cycle all incident edges are directed, while for nodes inside C no edge is directed yet. Nodes on C are candidate for processing. We will give a more detailed invariant later. The algorithm works as follows: we ini-



(a) Initial state (b) Intermediate state

FIGURE 4.2.2. Algorithm states

tialize the cycle C with all the interior neighbours of the exterior vertex c plus the other exterior vertices a and b. See Fig. 4.2.2(a). All edges between c and its interior neighbors will be directed toward c and be colored RED. (It is clear from Proposition 1.1.1 that we have to choose this direction. Furthermore we know that a Schnyder labelling also is a 3-tree-decomposition which implies that the edges must have the same color.) On this cycle C, every node has two neighbors: one in counterclockwise direction, called its *left* neighbor, and a *right* neighbor in clockwise direction.

From now on, we will maintain the following invariants:

- for every interior node *outside* C all incident edges will be colored according to Def. 1.2.1,
- no edge *inside* or *on* C is colored,

- every node *on* C, except the exterior nodes a and b, has an outgoing RED edge. All edges between this RED edge and the edge to v's left neighbor are incoming BLUE edges[1]. Accordingly, all edges between the RED edge and the edge to v's right neighbor are incoming and GREEN. (see Fig. 4.2.4(a))

Using these invariants, it will later be easy to verify the correctness of our algorithm.

We will now repeatedly "remove" nodes from C, i.e. afterwards it will be outside C, and replace them by nodes from inside C. A node on C is eligible for removal if its not adjacent to a *chordal* edge in C, that is, an edge between to nodes on C that are not neighbors on C, see Fig. 4.2.3. It is not allowed to remove one of the exterior nodes a and b.

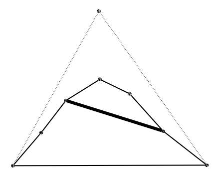
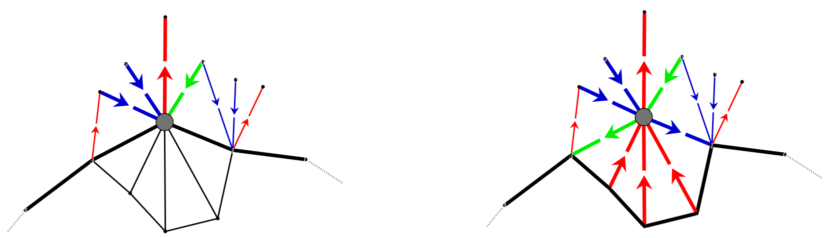


FIGURE 4.2.3.

The removal of node v then is done as follows (see Fig. 4.2.4):

- color the edge to v's left neighbor GREEN and direct it toward that neighbor,
- color the edge to v's right neighbor BLUE and direct it toward that neighbor,
- for all neighbors of v which are inside C, direct the connecting edges toward v and color them RED. In C, replace v by those nodes in order. (see Fig. 4.2.4(b))



(a) before removal

(b) after removal

FIGURE 4.2.4. Removing a node from C

One can verify that — given the invariants hold before the removal of v from C — v afterwards has outdegree 3 and its edges are colored

[1] *between* in this context means: start at the red edge and walk counterclockwise toward the left neighbor resp. clockwise toward the right neighbor. Note that any other interpretation would contradict the invariant that no edge on or in C is colored.

correctly. It is also easy to see that after v's removal the invariants still hold: the former left neighbor of v has got a new incoming GREEN edge, correctly placed between its outgoing RED edge and the edge to v. An according statement is true for v's former right neighbor. Finally, the nodes that have been inserted into C at this step have exactly one outgoing RED edge, while v's edge are all colored and it is now outside C.

To prove that our algorithm does indeed construct a Schnyder labelling for any triangular graph, it remains to show that we will always find a node $v \notin \{a, b\}$ that we can remove from C. However, it is not hard to show that in any cycle C there are nodes without chordal edges and it is only a little harder to show that there is one not contained in $\{a, b\}$ (for a complete proof see Appendix .1.) Thus, the algorithm runs until only a and b are left on C and then terminates with a Schnyder labelled graph.

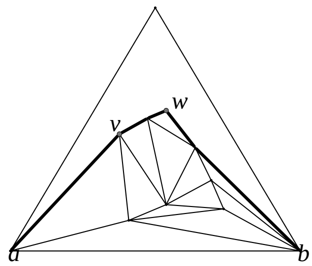
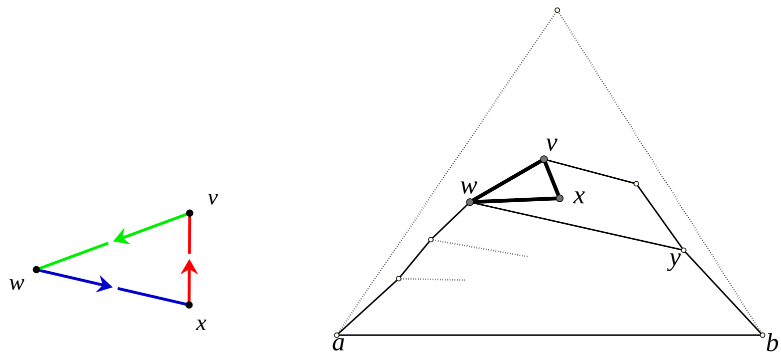


FIGURE 4.2.5. Path from a to b

4.2.1. Choosing Removable Nodes. In general there will be more than one removable node on C. We can control which Schnyder labelling is constructed by using different strategies how to choose the node to remove next. In particular, we can easily construct the Schnyder labellings without clockwise resp. counterclockwise cycles (in terms of 3-orientations: $\mathcal{L}_G$ and $\mathcal{R}_G$), by always choosing the "leftmost" resp. "rightmost" node. We already defined for each node on C a left and a right neighbor. If we drop the edge ab from C we get a path (see Fig 4.2.5) that runs from a on the left to b on the right, and the meaning of leftmost and rightmost is well-defined. In Fig. 4.2.5 v is the leftmost removable node and w is the rightmost removable node (there is a node right of w, but it is adjacent to a chordal edge). Some notation: if v and w are two nodes on the path from a to b, and v is closer to a then w, then $v \dots w$ denotes the subpath induced by v, w and the nodes between them. We will give a proof now why always choosing the leftmost node will result in a Schnyder labelling without ccw-cycles.

We know from Lemma 1.5.1 that we only have to make sure that the resulting Schnyder labelling does not contain ccw-triangles. So let us assume that we always chose the leftmost possible node and nevertheless constructed a ccw-triangle (see Fig. 4.2.6(a).) From the colors of the edges we can tell something about the order in which the algorithm must have processed the nodes v, w, and x (by *processed* we mean "to get removed from C", which obviously happens exactly once for each interior node). If two nodes are neighbors on C, then the edge between them will become either GREEN or BLUE, depending on which node will be removed first, and it will always become an outgoing edge of that node. This reasoning gives that v and w were once neighbors on C and v has been removed first, the same holds for w and x. So the order in which they were processed is v, w, x. (The RED edge from x to v tells us that v must have been processed before x, too.)



(a) ccw oriented triangle

(b) w is not removable

FIGURE 4.2.6

The crucial point is the color and direction of the edge vw: when v and w were neighbors on C, then — despite the fact that v is right of w and we are always choosing the leftmost possible node — v has been processed *before* w. This implies that w cannot have been removable, thus it must have been adjacent to a chordal edge wy inside C, see Fig. 4.2.6(b).The node y can not be left of w on C because then we would find a removable node in $y..w$ (with the same argument we used above: $y \dots w$ together with wy is a cycle). Hence y is right of v and planarity demands that x is inside the cycle formed by the path $w \dots y$ together with wy. The situation then looks like shown in Fig. 4.2.6(b). The removal of v will not make any node left of w removable, and the same argument as above implies that we always find a removable node between w and y. Therefore, before w itself can become removable, all nodes in the area defined by $w \dots y$ together with wy have to be removed. However, this implies that x is removed *before* w — a contradiction, since we saw that v, w, and x will be processed in that order. Note that in this argument we never assumed that the triangle formed by v, w, and x is a face boundary, so it is not restricted to 4-connected graphs.

COROLLARY 4.2.1. *Let G be a triangular graph. Apply the described algorithm and always choose the leftmost removable node. Then it terminates with a Schnyder labelling of G without counterclockwise oriented triangles, thus the 3-orientation induced by it is $\mathcal{L}_G$. Analogously, if you always choose the rightmost removable node, you will obtain $\mathcal{R}_G$.*[2]

Finally, we will argue that our algorithm is indeed only a variant of Schnyder's algorithm. In fact, it simulates the original algorithm without doing actual contractions. The basic construction we used for finding the colors and directions of edges is not new. Consider an ordering of the interior nodes $v_1, \dots, v_{n-3}$ in which v_1 is initially removable (with respect to some fixed exterior node c) and v_{i+1} is removable after $v_1 \dots v_i$ have been removed. Such a sequence is known as a *canonical labelling* of the interior nodes **[3]**, and Schnyder stated that each sequence in which his algorithm processes the nodes is a canonical labelling and vice versa. This implies that our algorithm is also capable of producing all possible Schnyder labellings on a planar graph.

The runtime of the algorithm is linear if we make sure that we don't spend too much time on maintaining the information about C and the removable nodes. However, this can be achieved by some bookkeeping. For every node v in G introduce a variable to keep track of how many of its neighbors are on C, and a boolean variable to see if v itself is on C. Thus we then can easily decide whether a node is removable, by verifiying that it has exactly two neighbors on C (and is itself on C, of course). These variables can be initialized in $O(|V| + 2|E|) = O(|V|)$. To maintain these variables, note that when a node gets first *onto* C, we have to increase its neighbors counters, and when removing it again from C, we have to decrease them. This amounts to $O(2|E|)$ altogether.

Finally we need to decide quickly which node is leftmost. To do this, we can keep the *removable* nodes in an ordered list, s.th. they appear in the same order as they are on C, so the first entry is always the leftmost one. To update this structure, we have to replace the removed node by those among its neighbors which are themselves now removable, which is easily done in a list. It might also happen that we have to unlink nodes from this list again, even if they are not actually removed from C, see Fig. **??**. This occurs when removing a node makes an edge a chordal edge. In Fig. **??** for example, after the removal of v, w will no longer be removable, because the edge xw is then chordal). This

[2] If you wondered in the previous chapters why we denoted the cycle-free 3-orientations by $\mathcal{L}_G$ and $\mathcal{R}_G$ — they stand for l̲eftmost resp. r̲ightmost

however will happen at most once for each edge. So the time we need dfor maintaining this list is again in $O(|E|)$, which leaves us with a total of $O(|V|)$.

CHAPTER 5

Exploring Schnyder Labellings

"Lange Zeit bin ich früh schlafen gegangen."
M. Proust
Auf der Suche nach der verlorenen Zeit

The last part of this work will describe a graphical application which allows you to create and view Schnyder labellings (resp. 3-orientation). It is based on LEDA[1] (see [**6**]), a library that allows to comfortably work with graphs and lets the user manipulate them visually. In particular, the main application frame (see Fig. 5.0.1) is a standard LEDAcomponent and thus documented elsewhere, for example in [**7**]. We will give a short introduction only and then describe our additional features.

5.1. What You Can Actually Do

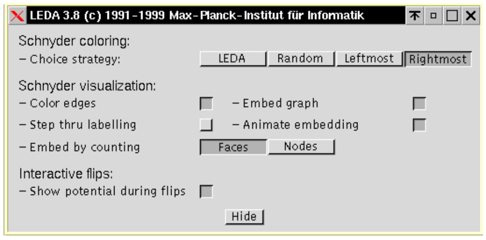
The basic LEDA-GraphWin consists mainly of a main pane in which you can manipulate the graph. You can add nodes by clicking and move them arouund by dragging. Edges can be created by clicking on the source node and then on the target node, the edge will show up as a rubberline while you choose its target. Nodes and edges can be deleted by clicking on them with the secondary (usually the *right*) mouse button, this will bring up a popup menu from which you can choose to delete them. By deleting a node you will delete all adjacent edges, too.

The main menu bar should be self-explaining. You will find familiar functions like **Save** or **Load**, and specific functions for creating, testing and embedding graphs. On the right end of the menu bar you will find two arrows. With these you can **undo** and **redo** your last modifications of the graph. The **done**-button on the far right end will be used for different purposes. In the standard mode it will close the application, after showing a confirmation dialog.

We added some functions to create and examine Schnyder labellings resp. 3-orientations. Usually we will speak of Schnyder labellings only,

[1] Library of Efficient Data types and Algortihms.

but we know from Section 1.2 that both concepts are equivalent, and Schnyder labellings are visually more appealing, anyway. You can find the controls to access these functions below the main menu. The functionality can be divided into two parts: generation of graphs and creation/visualization of Schnyder labellings and the distributive lattice of Schnyder labellings resp. 3-orientations.



5.1.1. Schnyder labellings.

3-Orient

The main operation is of course the creation of a Schnyder labelling. This is done via the button **3-Orient**. This will create a Schnyder labelling of the displayed graph. You will find the shown panel in a spearate window to control how the Schnyder labelling is created resp. displayed.

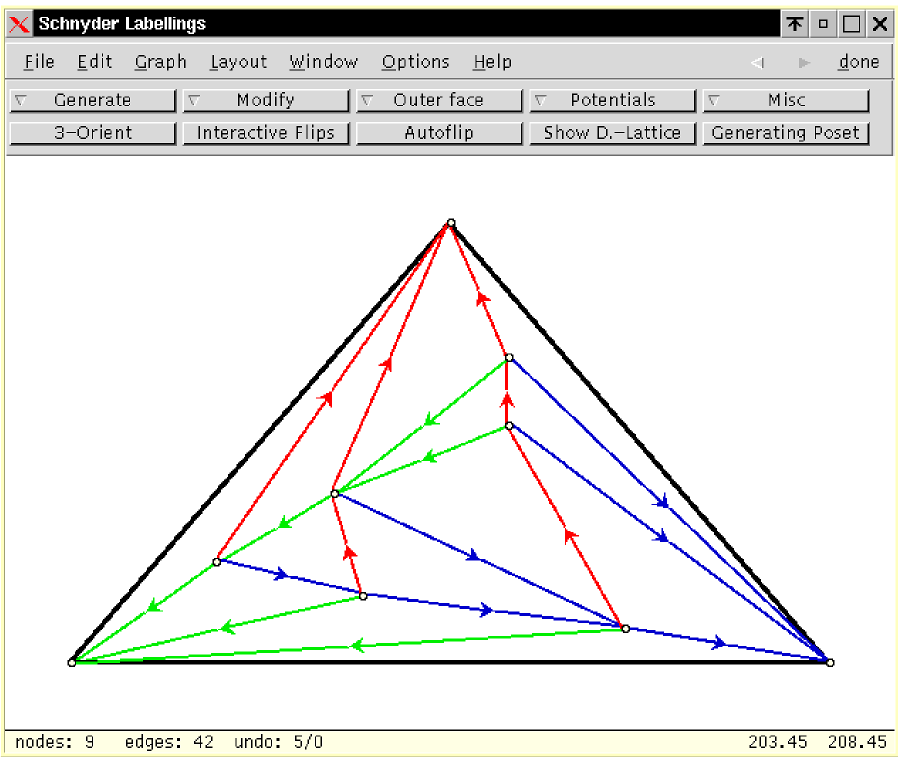


FIGURE 5.0.1.

leftmost
rightmost
random
LEDA
Color edges
Step thru labelling

The first four buttons allow to specify how the algorithm chooses the next removable node (see Sec. 4.2). **Leftmost**, **Rightmost**, and **Random** should be clear, while **LEDA** stands for a library function that chooses a node out of a set, without specifiying how this is done. The next options control how the Schnyder labelling is presented. With the **Color edges** checkbox you can switch off edge color, so that you are left with pure 3-orientations. **Step thru labelling** will allow you to see how the algorithm works: everytime a node is *removed* the color of its incident edges will be shown, and the algorithm will wait until you press **Done** before it continues. This option does not make too much sense without **Color edges** switched on.

Embed graph
Animate embedding
Show Options

If **Embed graph** is selected, a straight line embedding of the graph will be constructed after the Schnyder labelling has been created. This embedding is based on the actual Schnyder labelling and it is described in [**8**] how it can be constructed. The basic idea is to count **Faces** or **Nodes** for every node in the regions enclosed its three unique paths (in RED,GREEN, and BLUE). These two choices deliver slightly different embeddings[2]. **Animate embedding** finally does what it says: the nodes will be smoothly moved to their new positions. If **Embed graph** is switched off, you will not see any effect. Finally, the **Hide** button closes the option panel. You can re-open it with the menu item **Misc:Show Options**.

Flip outer face

The algorithm for constructing the Schnyder algorithm needs an outer face to start with. In triangular graphs all faces are equivalent, so we could start with an aribtrary one. You will see that after embedding the graph, the face will indeed be the outer face. But you can also choose to specify another outer face, e.g. when you constructed your own graph and want to make sure the face you consider to be outside will stay outside. This can be in various way, all of which expect you to specify *one* edge left of the desired face[3]. The quickest way to do this is to click on the edge with the left mouse button while pressing the CTRL-key. You will see that the edges bounding the corresponding face will turn red, while all other edges turn black. If for some reason you wanted the face on the other side of the edge, choose the menu item **Outer Face:Flip Outer Face.** This has nothing to do with face flips, but will simply make the opposite face be the outer face. In the

[2] FTWK: Note that we do not display the grid embeddings.

[3] A note on graphs in LEDA: our algorithm uses *bidirected* graphs. This means every edge uv appears twice, once directed from u to $\underline{v}$ and once from v to u. The advantage is that this way an edge can be uniquely associated with a face, in this case with the face left of it. This means that to specify a face you have to choose an edge that has the desired face *left of* it. The obvious disadvantage is that you have too many edges on the screen and that you have to make sure that graphs you constructed youself have this property, too. But the application usually will insert missing edges automatically, or you can do it manually with **Modify:Triangulate**.

Set outer face

Outer Face menu you will find one more item: **Set outer face**. This is the second way to do this. Select this item and you will be asked to select an edge. The third way is to use the edge context menu that you get by clicking with the right mouse button on an edge.

5.1.2. Flips. Since the first three chapters of this work are concerned a lot with face flips, you are also able to flip and flop faces yourself by choosing **Interactive flip.** You will then see that a little red square is placed in each flippable face and a yellow square in each floppable face. By actually clicking these squares you will flip or flop the resp. face. The view is updated to the settings in the option panel, for example if you chose **Animate embedding**, then after selecting a square a new embedding will be calculated and nodes will move to their new positions. If you have the option **Show potential during flips** activated, you will also see the value of the induced potential in each face. You can terminate the interactive flipping of faces by clicking with the right mouse button.

Interactive flip

Show potential during flips

If you don't like to flip faces yourself you can also let the program do this for you. **Autoflip** will repeatedly choose a flippable face and flip it. Again, the display is updated according to your settings. For maximum effect, you should create a rightmost Schnyder labelling first and select **Animate embedding.** But note that depending on the speed of your computer **Autoflip** might take its time. You can stop it by repeatedly clicking a mouse button in the main window[4] .

Autoflip

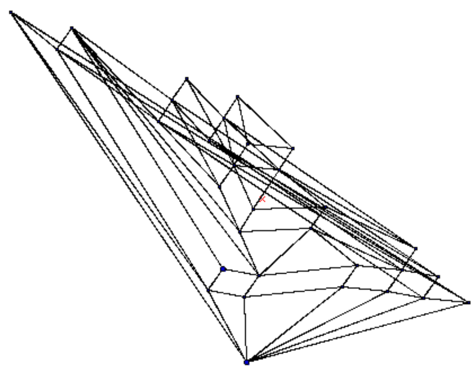


FIGURE 5.1.1. 3D View of an induced potential

5.1.3. Potentials. You have already seen one way to display the induced potential — during the interactive flipping. However, you can also explicitly have the induced potential displayed for a given Schnyder

[4]Actually you have to try to click the mouse button right between to flips, but experience has shown that repeated mouseclick work better. Try to click with the right or middle mouse button, otherwise you might by accident create some new nodes after **Autoflip** has stopped. We are aware that this is not very comfortable.

labelling. Simply choose **Induced Potential:Show induced potential** from the menu. Every face will be labelled with its potential value. Show induced potential

The program waits until you press **Done**, then it removes the labels again and you can continue. In the same menu, **Induced potential**, you will find another item, called **3D View**. Since you can imagine a 3D View

potential as a function that puts each face on a certain level, we implemented this function to let you visualize exactly that. You will get a 3D-wireframe model of a graph, such that each face is lifted to a level corresponding to its potential value, see Fig. 5.1.1. You can rotate it on all axis' by moving the mouse, and zoom in and out with the left resp. middle mouse button. Clicking the right mouse button will return to the standard display.

5.1.4. The Distributive Lattice Display. Right of the **Autoflip** button you find the button labelled **Show D.-Lattice.** This will Show D.-Lattice

construct the distribute lattice that is formed by *all* Schnyder labellings of the displayed graph. Independent of the settings in the option panel, a rightmost Schnyder labelling will be constructed and displayed and a second window will be opened which shows the lattice.

Each edge in the distributive lattice corresponds to a face flip/flop, that connects to 3-orientations. Accordingly, you will find the edges labelled with numbers which also appear as face labels in the graph displayed in the main window. You can verify that all edges at the maximal element correspond to flippable faces in the graph. The window containing the distributive lattice is closed by pressing **Done** in its upper right corner. The main frame will be deactivated as long as the second window is open.

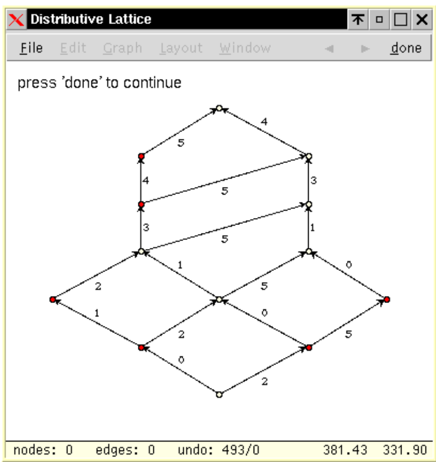


FIGURE 5.1.2. The view of a distributive lattice of a graph.

Unfortunately, this distributive lattice is constructed by actually doing all possible combinations of *face* flips on the current graph, which has two consequences. The first is that, depending on the size of the graph, it might take some time to construct the lattice, the second is that it only works for *4-connected* graphs. If you select this function for a non-4-connected graph, you will be presented with a proof that your graph is indeed not 4-connected, that is, a separating triangle will be displayed. In a dialog box you then have the choice to cancel the operation or make your graph 4-connected by removing some nodes (see below).

Generating Poset

5.1.5. The Generating Poset. The last button in this row is **Generating Poset.** This will construct the namely structure of the distributive lattice of all 3-orientations of the current graph and display it in a separate window. It it constructed from the distributive lattice L (so it as well might take some time) by the way described e.g. in[**9**]: take from L the subposet defined by the join-irreducible elements. Those elements are defined by the property that they cover exactly one element.

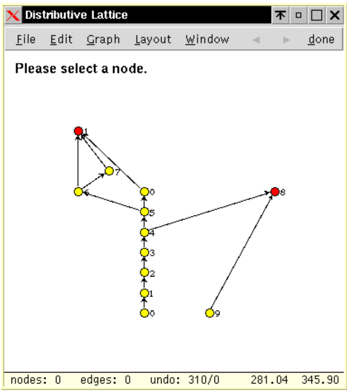


FIGURE 5.1.3. Generating poset

In our terms this means a 3-orientation that has exactly one flippable face. We will associate each each join-irreducible with this resp. face. You can see this association from the labels of the nodes in the separate window and the face labels in the main window. The maximal nodes in the generating poset are red, if you click on it, the corresponding face will be flipped, if possible and the node will be deleted. (We conjecture that the flippable faces are exactly the maximal nodes in the generating poset resp. in the poset obtained by deleting maximal nodes. This proof is still to be done.)

Check 4-connexity

Make 4-connected

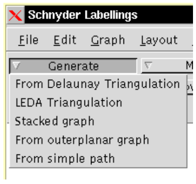
5.1.6. Graph Modification. You can also check whether your graph is actually 4-connected before you opt to construct its distributive lattice, by selecting **Check 4-connexity** from the **Misc** menu. In the case your graph is not 4-connected you will see, as stated above, a colored separating triangle. You then have the choice to fix this triangle, which means to check whether there are less nodes inside or outside the triangle and then remove these nodes. Note that the graph might still contain separating triangles afterwards. You can now repeat this process until no separating triangles are found, or you select **Make 4-connected** from the **Modify** menu, which will silently do this for you. All you will see is a short message how many nodes overall were removed from the graph.

Triangulate

There are some more operations in this menu: **Triangulate** adds edges to make the current graph a triangular graph (this function will in most cases silently be executed before anything else is done since we always need a triangular graph). This function also makes all edges of the graph bidirected: simply construct your triangular graph with single edges and then use **Triangulate** to add the missing edges. After reading Section 1.4 **Remove stacked nodes** should be clear. After executing this function, the graph will have no stacked node anymore. For the last choice in this menu, **Flip edges**, see below when we describe the graph generators.

Remove stacked nodes

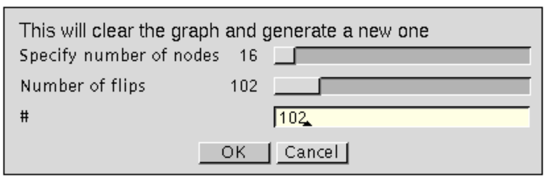
5.1.7. Generating Graphs. There are different kind of graph generators already built-in in the LEDApackage which you find under **Graph:Create.** We added some more which you will find in the **Generate** menu. When ever you choose to generate a graph you will be asked how many nodes you want in the graph. You can specify this via a textfield or by using the mouse on a slider. There is no generator for 4-connected graphs, so you will afterwards have to make them 4-connected yourself via **"Make 4-connected"** menu item[5]. You can generate the following types of triangular graphs



From Delaunay Triangulation: This will create randomly points in the unit square, add three additional points s.th. the convex hull will be a triangle and then constructs a graph from a Delaunay triangulation of this pointset. The graph you get will already be embedded. These graphs are often 4-connected, and if they are not, you can usually make them 4-connected by removing a few nodes only. The best choice to get to 4-connected graphs quickly.

LEDA Triangulation: This is a generator based on a apparently different triangulation algorithm in the LEDA library. These graphs usually are not 4-connected and loose more nodes than those from the Delaunay triangulation when making them 4-connected. Since these graphs appeared not very random, we offer a dialog to flip some faces (where flip in this case means a quadriliteral flip: the inner chord of a 4-cycle will be flipped, see Fig. 5.1.4). This might "randomize" the graph a bit.You can also explicitly call the **Flip edges** function from the **Modify** menu.

Flip edges



Stacked graph: See 1.4.1.

From outerplanar graph: This is a special case of a stacked graph. It constructs an outerplanar graph and triangulates the

[5]Remember that the operations that need a 4-connected graph will offer this function automatically.

outer face by adding a node in it an connecting all other node with this new node.

From simple path: Another special case of stacked graph: this will construct a path, add one node left of it an another one right of it, and connect both nodes with all nodes on the path. Finally the two additional nodes will be connected to make the graph triangular.

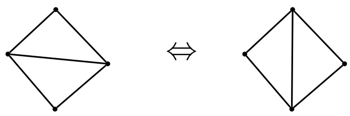


FIGURE 5.1.4. Quadriliteral flip

Fit graph in window

Reset edge colors

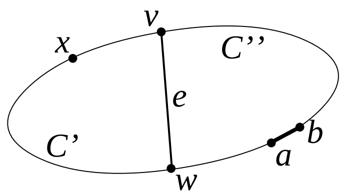
5.1.8. What is left. There are two remaining items in the **Misc** menu, which are not yet explained. For one this is **Fit graph in window.** This will adjust the graph layout such that it just fits into the visible area. The second is **Reset edge colors**, which whill exactly do that: all edges turn black except those of the current outer face: these will be colored red. Additionally this will make the program forget the current Schnyder labelling.

Appendix

.1. The Lost Proofs

LEMMA. *Let C be a cycle in an embedded planar graph, a and b to adjacent nodes on this cycle. Then there exists a node $x \notin \{a, b\}$ on C that is not adjacent to a chordal edge.*

PROOF. By induction on $|C|$: if $|C| = 3$, then there can be no chordal edges, so the third node on C will do. Let now $|C| = n + 1$. If C does not contain any chordal edges, then we are done. Otherwise let e be a chordal edge between some nodes v and w.



This edge e splits C into two new cycles C' and C'' (see figure), both containing e. W.l.o.g. the edge ab is not contained in C'. Applying the induction hypothesis to C' and the nodes v and w yields a node $x \notin \{v, w\}$ without chordal edges in C'. But if x has no chordal in C' it will not have any in C and it obviously is not in $\{a, b\}$.

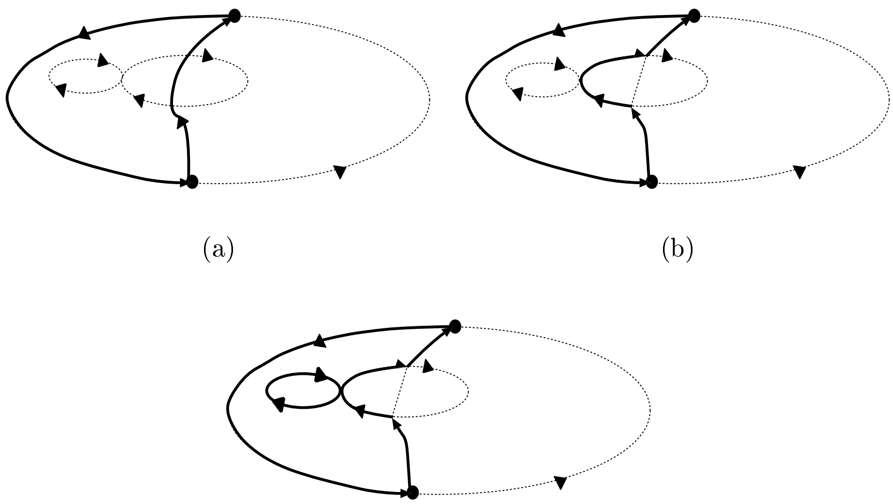
$\square$

DEFINITION .1.1. A region R is *ccw-bounded*, if whenever two faces f,g are adjacent and f is inside R and g outside, then f is left of g.

PROPOSITION .1.2. *In a 4-connected triangular graph, every ccw-bounded region contains a flippable face.*

PROOF. It is easy to see that a ccw-bounded region has a simple ccw-cycle as its outer boundary and each hole is bounded by a simple cw-cycle. These cw-cycles might touch each other or the outer boundary, see Fig. 1.1.2. The proof is by induction on the number of contained faces. If this is one, then R is a flippable face and we are done. We claim that if R consist of more faces we can always find a ccw-bounded region entirely contained in R'. So let R contain more than one face. Its outer boundary then can no longer be a triangle because this would be a separating triangle.

Let C be the *outer* boundary of R. Consider the region $fill(R)$, which is the region bounded by C — in other words, we are filling all holes in R. This is a simple region, whose boundary is not a triangle. So by Lemma 1.1.2, we know there must be an interior edge of $fill(R)$ leaving a node on its boundary, which can be extended to a path P crossing $fill(R)$ (see Fig. 1.5.1), possibly crossing holes, too. Now P together with one segment C' of C forms a new ccw-cycle PC'. The interior of this cycle is a region that contains faces from R and possibly faces from holes. We will now add again all holes to this region, one by one. At each step we will update the boundary, so that in the end we are left with a ccw-bounded region that contains a proper subset of faces of R.



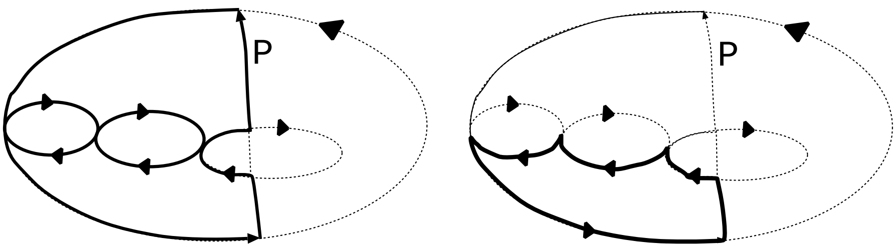
(a) (b)

(c)

FIGURE .1.1.

How do we remove the holes? If a hole does not intersect with the current region at all the boundary does not change. If it crosses the current boundary we update it as shown in Fig. .1.1(a) and .1.1(b)again have a region that is ccw-bounded. If a hole touches only from the inside, we again have nothing to do, see Fig. .1.1(b) and .1.1(c). After removing all holes, every face inside our region is a face from R, and it is ccw-bounded, but maybe we are left with a region that no longer is edge-connected, see Fig. .1.2(a). However, in this case we can split off a smaller subregion, that again is ccw-bounded and edge-connected. So we found a ccw-bound region, containing less faces than R. By induction hypothesis, in this region we will find a flippable face f, which then also is in R.

□



(a) (b)

FIGURE .1.2

Bibliography

[1] H. de Fraysseix and P. Ossona de Mendez. Connectivity of planar graphs. Technical Report 173, CAMS, 1999.

[2] H. de Fraysseix and P. Ossona de Mendez. On topological aspects of orientations. *Proc. of the 5. Czech-Slowak Symposium on Combinatorics, Graph Theory, Algorithms and Applications, Discr. Mathematics*, to appear.

[3] H. de Fraysseix, J.Pach, and R. Pollack. Small sets supporting Fáry embeddings of planar graphs. In *Proc. 20th Ann. ACM Symp. on Theory of Computing*, pages 426–433, 1988.

[4] P. Ossona de Mendez. *Orientations bipolaires*. PhD thesis, Ecole des Hautes Etudes en Sciences Sociales, Paris, 1994.

[5] Reinhard Diestel. *Graphentheorie*. Springer Verlag, Berlin Heidelberg, 1996.

[6] K. Mehlhorn, S. Näher, M. Seel, and C. Uhrig. LEDA - A Library of Efficient Data Types and Algorithms. `http://www.mpi-sb.mpg.de/LEDA`.

[7] K. Mehlhorn and St. Näher. *LEDA. A Platform of Combinatorial and Geometric Computing*. Cambridge University Press, 1999.

[8] Walter Schnyder. Embedding planar graphs on the grid. *Proc. 1st ACM-SIAM Sympos. Discrete Algorithms*, pages 138–148, 1990.

[9] Richard P. Stanley. *Enumerative Combinatorics*, volume I. Wadsworth & Brooks/Cole Advanced Books & Software, Monterey California, 1986.

"There was things he stretched,
but mainly he told the truth."

M. Twain
Huckleberry Finn