Upward Planarity Checking: "Faces Are More than Polygons"*
(Extended Abstract)

Giuseppe Di Battista[1] and Giuseppe Liotta[2]

[1] Dipartimento di Informatica e Automazione, Università di Roma Tre
via della Vasca Navale 79, 00146 Roma, Italy. gdb@dia.uniroma3.it
[2] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"
via Salaria 113, 00198 Roma, Italy. liotta@dis.uniroma1.it

Abstract. In this paper we look at upward planarity from a new perspective. Namely, we study the problem of checking whether a given drawing is upward planar. Our checker exploits the relationships between topology and geometry of upward planar drawings to verify the upward planarity of a significant family of drawings. The checker is simple and optimal both in terms of efficiency and in terms of degree.

1 Introduction

The intrinsic structural complexity of the implementation of geometric algorithms makes the problem of formally proving the correctness of the code unfeasible in most of the cases. This has been motivating the research on *checkers*. A checker is an algorithm that receives as input a geometric structure and a predicate stating a property that should hold for the structure. The task of the checker is to verify whether the structure satisfies or not the given property. Here, the expectation is that it is often easier to evaluate the quality of the output than the correctness of the software that produces it. Several authors [17, 16, 5] agree on the basic features that a "good" checker should have:

Correctness: The checker should be correct beyond any reasonable doubt. Otherwise, one would fall into the problem of checking the checker.
Simplicity: The implementation should be straightforward.
Efficiency: The expectation is to have a checker that is not less efficient than the algorithm that produces the geometric structure.
Robustness: The checker should be able to handle degenerate configurations of the input and should not be affected by errors in the flow of control due to round-off approximations.

* Research supported in part by the ESPRIT LTR Project no. 20244 - ALCOM-IT and by the CNR Project "Geometria Computazionale Robusta con Applicazioni alla Grafica ed al CAD."

S.H. Whitesides (Ed.): GD'98, LNCS 1547, pp. 72–86, 1998.

Checking is especially relevant in the graph drawing context. In fact, graph drawing algorithms are among the most sophisticated of the entire computational geometry field, and their goal is to construct complex geometric structures with specific properties. Also, because of their immediate impact on application areas, graph drawing algorithms are usually implemented right after they have been devised. Further, such implementations are often available on the Web without any certification of their correctness. Of course, the checking problem becomes crucial when the drawing algorithm deals with very large data sets, when a simple complete visual inspection of the drawing is difficult or unfeasible.

Devising graph drawing checkers involves answering only apparently innocent questions like: "is this drawing planar?" or "is this drawing upward?" or "are the faces convex polygons?". The problem of checking the planarity of a subdivision has been pioneered in [18, 5]. In those papers linear time algorithms are given to check the planarity of a subdivision composed by convex faces. The inputs are the subdivision plus its topological embedding in terms of the ordered adjacency lists of the edges. Unfortunately, extending the above techniques to checking the planarity of a subdivision whose faces are not constrained to be convex, relies on the usage of algorithms for testing the simplicity of a polygon. The only general linear time algorithm known for this problem is the fairly complex algorithm in [3]. Hence, devising a checker based on such algorithm would not satisfy the simplicity requirement. The algorithm in [3] tests the simplicity of a polygon by means of an intermediate triangulation step. Alternative algorithms that can triangulate in linear time special classes of polygons have been devised. See e.g. [10, 8]. Other almost optimal algorithms can be found in [12, 4, 20].

In this paper we study the problem of checking the *upward planarity* of a drawing. Upward planarity is a classical topic in graph drawing and several papers deal with the problem of testing whether a given graph has an upward planar drawing and eventually constructing it. For an overview, see [6]. We look at the problem from a different perspective. The main results of this paper are:

(i) We introduce and study *regular* upward planar embeddings. We show that such embeddings coincide with those that have a "unique" including planar st-digraph. There are several families of digraphs whose upward planar embeddings are always regular. E.g. rooted trees, planar st-digraphs, and planar sT-digraphs (i.e. single-source digraphs). The great majority of algorithms for constructing upward planar drawings receive as input such digraphs. (ii) We exploit the concept of regularity to investigate the relationships between topology and geometry of upward planar drawings. In particular, we show that an upward drawing of a regular planar upward embedding satisfies strong constraints on the left-to-right ordering of the edges. (ii) Based upon the above results and under the assumption of regularity we present a linear time checker to test whether a given drawing Γ is upward planar. Our checker receives as input the set of vertices and bends of Γ (represented as pairs of integer coordinates), the set of oriented edges of Γ, and the embedding of Γ, i.e. the circular ordering of the edges incident on each vertex of Γ. An example of a drawing whose upward planarity can be checked by our algorithm is shown in Figure 1. (iii) We further analyze the effectiveness of our checker by adopting the notion of *degree* which takes into account the arithmetic precision required by the checker to carry out error-free computations. We show that our checker has (optimal) degree 2.

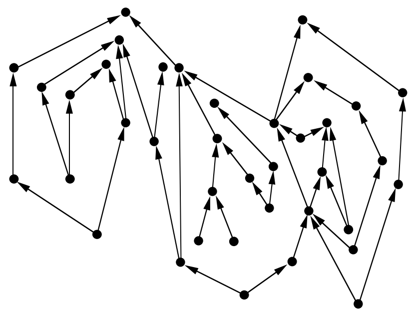

Fig. 1. Example of drawing whose upward planarity can be checked with the algorithm presented in this paper.

Our techniques do not exploit the polygon simplicity perspective but rely on the relationships between topology and geometry. Namely, in general triangulating the faces of a drawing does not appear to be strictly necessary when testing its planarity; also, the faces of an upward drawing can be very different from those polygons for which a simple triangulation algorithm is known.

2 Preliminaries

We first recall the notion of algorithmic degree, then we recall basic definitions and properties of upward planar drawings and embeddings. We assume familiarity with basic graph and geometric terminology. See also [11, 19].

The notion of *degree* as a measure of the precision that can be required by an error-free implementation of an algorithm has been introduced in [14, 15]. We briefly recall some terminology and results.

The numerical computations of a geometric algorithm are basically of two types: tests (predicates) and constructions. Tests are associated with branching decisions in the algorithm that determine the flow of control. Constructions are needed to produce the output data of the algorithm. While approximations in the execution of constructions are acceptable, provided that the error magnitude does not exceed the resolution required by the application, approximations in the execution of tests may produce an incorrect branching of the algorithm, thus giving rise to *structurally* incorrect results. The exact-computation paradigm therefore requires that tests be executed with total accuracy.

Geometric and graph drawing algorithms can be therefore analyzed on the basis of the complexity of their test computations. Any such computation consists of evaluating the sign of an algebraic expression over the input variables, constructed using an adequate set of operators, such has $\{+,-,\times,\div,\sqrt[2]{}\}$. This can be reduced to the evaluation of the signs of multivariate polynomials derived from the expression.

A primitive variable is an input variable of the algorithm and has conventional arithmetic degree 1. The arithmetic degree of a polynomial expression E is the common arithmetic degree of its monomials. The arithmetic degree of a monomial is the sum of the arithmetic degrees of its variables. An algorithm has *degree* d if its test computations involve the evaluation of multivariate polynomials of arithmetic degree at most d. A problem has *degree* d if any algorithm that solves the problem has degree at least d.

A straightforward consequence of a result in [5] is the following.

Theorem 1. *The upward planarity checking problem has degree at least* 2.

We borrow some terminology and results from [7, 2]. Let G be a planar digraph. An *upward drawing* of G is a drawing such that all edges are represented by curves monotonically increasing in a common direction, for example the vertical one. A digraph that admits an upward planar drawing is *upward planar.*

An *st-digraph* is an acyclic digraph with exactly one source s and exactly one sink t and such that s and t are adjacent. An *sT-digraph* is an acyclic digraph with exactly one source s.

Lemma 1. [13, 7] *An upward planar digraph is a subgraph of a planar st-digraph.*

Let G be an embedded planar digraph. A vertex of G is *bimodal* if its incident list can be partitioned into two possibly empty linear lists, one consisting of incoming edges and the other consisting of outgoing edges. If all its vertices are bimodal then G and its embedding are called *bimodal.* A digraph is *bimodal* if it has a planar bimodal embedding.

Let f be a face of a bimodal digraph G. Visit the contour of f counterclockwise (i.e. such that the face remains always to the left during the visit). A vertex v of f with incident edges e_1 and e_2 is a *switch* if the the direction of e_1 is opposite to the direction of e_2 (note that e_1 and e_2 may coincide if the digraph is not biconnected). If e_1 and e_2 are both incoming (outgoing) v is a *sink switch* (*source switch*) of f. Let $2n_f$ be the number of switches of f. The *capacity* c_f of f is defined to be $n_f - 1$ if f is an internal face and $n_f + 1$ if f is the external face.

An assignment of the sources and sinks of G to its faces such that the following properties hold is *upward consistent*: (i) A source (sink) is assigned to exactly one of its incident faces. (ii) For each face f, the number of sources and sinks that are assigned to f is equal to c_f.

Theorem 2. [2] *Let G be an embedded bimodal digraph; G is upward planar if and only if it admits an upward-consistent assignment.*

Let G be an embedded bimodal digraph that has an upward-consistent assignment. According to Theorem 2, in [2] it is defined the concept of *upward planar embedding* of G as its planar embedding for which, for each face f, the switches of f are labeled S or L. A switch is labeled L if it is a source or a sink assigned to f, S otherwise. If f is internal, then the number of its switches labeled L is $c_f - 1$, else the number of its switches labeled L is $c_f + 1$. A face of G with the above properties is *upward consistent*. The circular list of labels of f will be usually called the *labeling* of f and denoted as σ_f. Also, S_{σ_f} (L_{σ_f}) denotes the number of S-labels (L-labels) of σ_f.

Observe that an embedded bimodal digraph can have many upward planar embeddings each corresponding to a certain upward-consistent assignment.

Property 1. **[2]** For an upward consistent internal (external) face f we have: $S_{\sigma_f} = L_{\sigma_f} + 2$ ($L_{\sigma_f} = S_{\sigma_f} + 2$).

An immediate consequence is the following.

Property 2. The labeling σ_f of an upward consistent internal (external) face f has at least two consecutive S-labels (L-labels).

Another consequence of the results in [2] is that, given a planar upward embedding of a digraph G it is always possible to construct a planar st-digraph including G by adding to G a new source s, a new sink t, edge (s,t), and a suitable set of (dummy) edges. Such edges connect either pairs of switches or external face switches with s or t. More formally, we can elaborate the concepts in [2] as follows. Given an upward planar embedded digraph G a *saturator* of G is a set of edges (each edge a *saturating edge*) plus two vertices s and t connected by edge (s,t). A saturating edge is such that:

- A saturating edge can either connect two switches of the same face, or it can connect a sink switch labeled L of the external face to t, or it can connect s to a source switch labeled L of the external face.
- For a saturating edge (u,v), $u,v \neq s,t$, either u is a source switch labeled S and v is a source switch labeled L or u is a sink switch labeled L and v is a sink switch labeled S. In the former case we say that u *saturates* v and in the latter case we say that v *saturates* u.
- The faces obtained with the insertion of a saturating edge are upward consistent.

Examples of upward planar embeddings can be found in Figure 2. The dashed edge of Figure 2(c) is a saturating edge.

The set of saturating edges added to a face is also called *saturator* of that face. We mainly focus on properties of saturators of internal faces. Analogous properties hold for the external face. The following property relates the labeling of an internal face f to the labeling of the faces obtained when inserting a saturating edge.

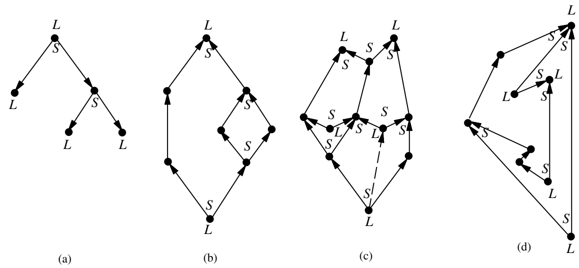

Fig. 2. Examples of labelings, faces, and digraphs: (a) A planar upward embedded rooted tree. (b) A planar upward embedded *st*-digraph. (c) A planar upward embedded *sT*-digraph. The dashed edge is a saturating edge. (d) A more complex planar upward embedded digraph.

Property 3. Let f be an internal face of an upward planar embedded digraph, let e be a saturating edge of f, and let $\sigma_f = \sigma_1 S \sigma_2 L$, where the S and L in evidence are the labels of the endvertices of e. Then, $S_{\sigma_1} = L_{\sigma_1} + 1$ and $S_{\sigma_2} = L_{\sigma_2} + 1$.

Proof. By the definition of saturator, both the faces resulting by the insertion of e are upward consistent. Observe that such faces have labeling $\sigma_1 S$ and $S \sigma_2$. Hence, by Property 1 we have that $S_{\sigma_1} = L_{\sigma_1} + 1$ and $S_{\sigma_2} = L_{\sigma_2} + 1$.

A saturator of G is said to be *complete* if for every face f and for every switch u of f labeled L, u is an endvertex of an edge of the saturator.

Property 4. Each upward planar digraph G is a subgraph of a planar *st*-digraph constructed by adding to G a complete saturator of an upward planar embedding of G.

Observe that an upward planar embedding can have, in general, many complete saturators.

Property 5. Each planar upward drawing of a digraph G is a subdrawing of a planar upward drawing of a planar *st*-digraph enclosing G and constructed by adding to G a complete saturator.

3 Regular Upward Embeddings

In this section we characterize the family of upward planar digraphs for which there exists a unique complete saturator. Also, we study the relationships between topological properties and upward drawings of such digraphs.

3.1 Regularity

Our characterization is based on a certain type of labeling. Namely, let G be an upward planar embedded digraph. An internal face f of G has a *regular labeling* if σ_f does not contain two distinct maximal subsequences σ_1 and σ_2 of S-labels such that $S_{\sigma_1} > 1$ and $S_{\sigma_2} > 1$. An external face f of G has a *regular labeling* if σ_f does not contain two consecutive S-labels. A face of G with a regular labeling is a *regular* face. For example all labelings of Figure 2 are regular.

Property 6. In the labeling of a regular internal face f with more than two switches, there is always a maximal subsequence of at least three consecutive S-labels.

Proof. By contradiction. Suppose that the labeling of the face f is $\sigma_f = \sigma_1 LSSL$. By Property 1, we have $S_{\sigma_1} = L_{\sigma_1} + 2$. Therefore, σ_1 has two consecutive S-labels which implies the non regularity of the face.

Based on the notion of regular face, a family of upward planar embedded digraphs can be defined. An upward planar embedded digraph is *regular* if all its faces have a regular labeling. The corresponding embedding is also called *regular.* An upward planar digraph is *regular* if all its upward planar embeddings are regular.

The following theorem lists important families of regular digraphs.

Theorem 3. *Rooted trees, planar st-digraphs, and upward planar sT-digraphs are regular.*

Proof. We prove the regularity by describing with regular expressions the labeling of the faces of any upward planar embedding. Rooted trees have one (external) face in any upward planar embedding. The face is regular since its labeling is $LL(LS)^*$. See, for example, Figure 2(a). Concerning planar st-digraphs: the external face is labeled LL and the internal faces are labeled SS. See, for example, Figure 2(b). Upward planar sT-digraphs are such that the external face is labeled $LL(LS)^*$ and the internal faces are labeled $SS(SL)^*$.

An example of a more complex regular upward embedded digraph, that does not fall in any of the classes mentioned in Theorem 3, is shown in Figure 2(d).

In Section 2 it has been observed that an upward planar embedding can have, in general, many complete saturators. The notion of regular embedding allows us to characterize those planar upward embeddings that have only one complete saturator. We start by studying how a switch labeled L can be saturated in a regular face.

Two switches of a face f of an upward planar embedded digraph are *neighbor switches* if their labels are in the same subsequence σ_1 of σ_f such that all labels in σ_1 are S (L) labels.

Lemma 2. *In a regular face f with more than two switches, if a switch u saturates a switch v, then u has at least one neighbor switch.*

Proof. Suppose the face is internal. By contradiction. Let the labeling of f be: $\sigma_f = \sigma_1 LSL\sigma_2 L$, where the last L-label is the label of v and the S-label is the label of u. After the insertion of edge (u, v) the two new faces f_1 and f_2 have labeling $\sigma_1 LS$ and $SL\sigma_2$, respectively. Also, by Property 2 σ_f has at least two consecutive S-labels. Suppose wlog that such consecutive S-labels are in σ_1. It follows, by the regularity of σ_f, that σ_2 does not have two consecutive S-labels. However, by Property 2 applied to face f_2 this is not possible.

Lemma 2 restricts the set of possible switches labeled S that can be used to saturate the switches labeled L in a regular face. The next lemma allows us to further restrict our attention to only one switch of such a set. The lemma holds for any (even non-regular) face of an upward planar digraph and will be used in its general form to prove a next theorem.

Lemma 3. *Let f be an internal face of an upward planar embedded digraph, such that f has more than two switches, and let u and v be two switches of f. If v can saturate u, then none of the neighbor switches of v can saturate u.*

Proof. Let $\sigma_f = \sigma_1\sigma_2 S\sigma_3\sigma_4 L$ be the labeling of f where: S is the label of v, L is the label of u, σ_2 and σ_3 are the subsequences of the S-labels of the neighbor switches of v (by Lemma 2 at most one of such subsequences may be empty), and σ_1 and σ_4 contain the remaining labels of σ_f. Clearly, $L_{\sigma_2} = L_{\sigma_3} = 0$. Therefore, since v can saturate u, by Property 3 we have that $S_{\sigma_1} + S_{\sigma_2} = L_{\sigma_1} + 1$. It follows that $S_{\sigma_2} = L_{\sigma_1} - S_{\sigma_1} + 1$. Observe that the position of the label of the switch saturating u in the sequence $\sigma_2 S\sigma_3$ is univocally determined by the values of L_{σ_1} and S_{σ_1}. Hence, there is only one switch of f that can saturate u among v and its neighbor switches.

We are now ready to prove one of the main results of this section.

Theorem 4. *An upward planar embedding has only one complete saturator if and only if it is regular.*

Proof. We concentrate on internal faces. Similar arguments hold for the external face. Let G be a digraph with a given regular upward planar embedding and let f be an internal face of G. Lemmas 2 and 3 imply that for every switch v labeled L of f there exists a unique switch u such that (u, v) is a saturating edge. Hence, if an upward planar embedding is regular, then there is only one complete saturator. To prove the necessity, we now show that if an internal face f of an upward planar embedded digraph G has a switch u labeled L and two distinct switches v and w such that both v and w can saturate u, then f (and hence G) is not regular. We have that $\sigma_f = \sigma_1 S\sigma_2 S\sigma_3 L$, where the first S is the label of v, the second S is the label of w, and L is the label of u. Since v (w) can saturate u, the saturating edge (u, v) ((u, w)) splits f into two upward consistent faces. Thus, by Property 1 there are two consecutive S labels in $\sigma_1 S$ ($S\sigma_3$). By Lemma 3 v and w are not neighbor switches. Hence, $L_{\sigma_2} \geq 1$. This implies the non-regularity of f.

The topology of an upward planar embedded digraph G induces ordering relationships on its edges, which correspond to a set of geometric constraints that have to be satisfied by an upward planar drawing of G. Together with regularity properties, the study of such ordering relationships has revealed to be a basic ingredient for the design of our efficient low-degree upward planarity checker.

3.2 Precedence and Dominance

Let G be a planar embedded st-digraph. Let e_1 and e_2 be two distinct edges of G. We say that e_1 is *to the left of* e_2, and denote it as $e_1 \overset{l}{\prec}_G e_2$, when:

1. there exists a drawing Γ of G and two distinct points $p_1 \in \Gamma(e_1)$ and $p_2 \in \Gamma(e_2)$ such that $y(p_1) = y(p_2)$ and $x(p_1) < x(p_2)$;
2. for any upward planar drawing Γ of G and for any two distinct points $p_1 \in \Gamma(e_1)$ and $p_2 \in \Gamma(e_2)$ such that $y(p_1) = y(p_2)$, we have that $x(p_1) < x(p_2)$.

The following properties can be easily proved.

Property 7. For each pair $(u_1, v_1), (u_2, v_2)$ of edges of G, the relationship $(u_1, v_1) \overset{l}{\prec}_G (u_2, v_2)$ holds if and only if: (1) it does not exist a directed path from s to t containing both edges, and (2) there exists a vertex w and two internally disjoint directed paths π_1 and π_2 such that π_1 contains (u_1, v_1), π_2 contains (u_2, v_2), π_1 and π_2 share w, and the edge of π_1 incident on w is to the left of the edge of π_2 incident on w in the ordering of the incoming edges incident on w.

Property 8. Let G be a planar embedded st-digraph. Relation $\overset{l}{\prec}_G$ is transitive.

We now define a precedence relationship between two edges of an upward planar embedded digraph. Let e_1 and e_2 be two distinct edges of an upward planar embedded digraph G. We say that e_1 is *to the left of* e_2, and denote it as $e_1 \overset{l}{\prec}_G e_2$, when for each including st-digraph G' obtained by adding a complete saturator to G we have that $e_1 \overset{l}{\prec}_{G'} e_2$.

From Property 8 it follows that:

Property 9. Let G be an upward planar embedded digraph. Relationship $\overset{l}{\prec}_G$ is transitive.

Let e_1 and e_2 be two distinct edges of an embedded planar st-digraph G. We say that e_1 is *dominates* e_2, and denote it as $e_1 \overset{u}{\prec}_G e_2$, when for any upward planar drawing Γ of G and for any two distinct points $p_1 \in \Gamma(e_1)$ and $p_2 \in \Gamma(e_2)$ we have that $y(p_1) > y(p_2)$.

Property 10. For each pair e_1, e_2 of distinct edges of G $e_1 \overset{u}{\prec}_G e_2$ if and only if there exists a directed path from s to t containing both e_1 and e_2 and such that when going from s to t along the path e_1 is encountered before e_2.

Property 11. Let G be a planar embedded st-digraph. Relation $\overset{u}{\prec}_G$ is transitive.

We now define a dominance relationship between two edges of an upward planar embedded digraph. Let e_1 and e_2 be two distinct edges of an upward planar embedded digraph G. We say that e_2 *dominates* e_1, and denote it as $e_1 \overset{u}{\prec}_G e_2$, when for each including st-digraph G' obtained by adding a complete saturator to G we have that $e_1 \overset{u}{\prec}_{G'} e_2$.

From Property 11 it follows that:

Property 12. Let G be an upward planar embedded digraph. Relationship $\overset{u}{\prec}_G$ is transitive.

Observe that since relationships $\overset{l}{\prec}_G$ and $\overset{u}{\prec}_G$ are mutually exclusive for an embedded planar st-digraph, the following property holds.

Property 13. Let G be an upward planar embedded digraph. Relationships $\overset{l}{\prec}_G$ and $\overset{u}{\prec}_G$ are mutually exclusive.

Also notice that, because of Property 5, the relationships $\overset{l}{\prec}_G$ and $\overset{u}{\prec}_G$ defined for an upward planar embedded digraph give left-to-right and up-down constraint that hold for any drawing of G.

We are now ready to characterize the upward planar embedded digraphs such that for any pair of edges, they are in the $\overset{u}{\prec}_G$ or in the $\overset{l}{\prec}_G$ relationship. From the above discussion it follows a sufficient condition:

Property 14. Upward planar embedded st-digraphs are such that for each pair e_1, e_2 of edges either $e_1 \overset{l}{\prec}_G e_2$ or $e_2 \overset{l}{\prec}_G e_1$ or $e_1 \overset{u}{\prec}_G e_2$ or $e_2 \overset{u}{\prec}_G e_1$.

A complete characterization is given in the following theorem.

Theorem 5. *Let G be an upward planar embedded digraph. For each pair e_1, e_2 of edges of G it holds that either $e_1 \overset{l}{\prec}_G e_2$, or $e_2 \overset{l}{\prec}_G e_1$, or $e_1 \overset{u}{\prec}_G e_2$, or $e_2 \overset{u}{\prec}_G e_1$, if and only if G is regular.*

Proof. First, we prove the sufficiency. Namely, we prove that if G is regular then for each pair e_1, e_2 of edges of G it holds that either $e_1 \overset{l}{\prec}_G e_2$, or $e_2 \overset{l}{\prec}_G e_1$, or $e_1 \overset{u}{\prec}_G e_2$, or $e_2 \overset{u}{\prec}_G e_1$.

If G is regular then, by Property 4, and Theorem 4 it follows that there exists a unique planar embedded st-digraph G' enclosing G and constructed by adding a complete saturator to G. The sufficiency is immediately implied by Property 14.

Suppose now, for a contradiction, that there exists a non-regular upward planar embedded digraph G such that between any pair of edges of G either the $\overset{l}{\prec}_G$ or the $\overset{u}{\prec}_G$ relationship is defined. Let f be a non-regular face of G, let u be a sink switch of f labeled L, and let v and w be sink switches labeled S that can saturate u (the proof is symmetric for the case that u, v, and w are source switches). Let e_1 and e_2 be the two edges of f incident on v such that $e_1 \overset{l}{\prec}_G e_2$; let e_3 and e_4 be the two edges of f incident on w such that $e_3 \overset{l}{\prec}_G e_4$. Finally, let e_5 and e_6 be the two edges of f incident on u. Since both v and w can saturate u, consider two different planar embedded *st*-digraphs that include G: G' has the saturating edge (u, v), while G'' has the saturating edge (u, w).

In G', we have that $e_1 \overset{l}{\prec}_{G'} e_5 \overset{l}{\prec}_{G'} e_2$ by Property 7. Since for all pairs of edges of G either the $\overset{l}{\prec}_G$ or the $\overset{u}{\prec}_G$ relationship is defined and since there exists an *st*-digraph (constructed with a complete saturator) including G for which $e_5 \overset{l}{\prec}_{G'} e_2$, we can conclude that e_5 is to the left of e_2 also for G, i.e. $e_5 \overset{l}{\prec}_G e_2$.

In G'' we have that $e_3 \overset{l}{\prec}_{G''} e_5 \overset{l}{\prec}_{G''} e_4$. With analogous reasoning as above, we conclude that $e_3 \overset{l}{\prec}_G e_5$.

Now, four cases are possible: Either $e_2 \overset{l}{\prec}_G e_3$ or $e_3 \overset{u}{\prec}_G e_2$, or $e_3 \overset{l}{\prec}_G e_2$, or $e_2 \overset{u}{\prec}_G e_3$. We show a contradiction for the first two cases. The proof for the other cases is symmetric.

Namely, if $e_2 \overset{l}{\prec}_G e_3$, since we have shown that $e_5 \overset{l}{\prec}_G e_2$, by the transitivity property (Property9) it follows $e_5 \overset{l}{\prec}_G e_3$. But we should also have $e_3 \overset{l}{\prec}_G e_5$, a contradiction. If, in turn, $e_3 \overset{u}{\prec}_G e_2$, in G'' we have that $e_5 \overset{u}{\prec}_{G''} (u, w) \overset{u}{\prec}_{G''} e_3 \overset{u}{\prec}_{G''} e_2$ (Property10). By the transitivity property (Property11) it follows that in G'' $e_5 \overset{u}{\prec}_{G''} e_2$ which implies that also in G it should be $e_5 \overset{u}{\prec}_G e_2$. But, we should also have $e_5 \overset{l}{\prec}_G e_2$, a contradiction because of Property 13.

4 Upward Planarity Checking

Let Γ be a connected polygonal-line drawing of a digraph G with n vertices and bends. An *upward planarity checker* of Γ receives as input the set of vertices and bends of Γ represented as pairs of integer coordinates, the set of oriented edges of Γ, and the embedding of Γ, i.e. the circular ordering of the edges incident on each vertex of Γ.

Our upward planarity checker executes three tests in sequence. If a test fails, then the checker rejects Γ, otherwise it executes the test that follows in the sequence. At the end of the procedure, either a certificate for Γ is provided or a message that rejects Γ providing evidence of the property that is not respected by Γ. The tests performed by the checker are listed below.

`Embedding-Test:` Verify whether the given embedding is planar. This is equivalent to verifying whether there exists a drawing Γ' of G that preserves the given embedding and such that no two edges of Γ' cross. The bimodality of the given embedding is also verified.

`Upwardness-Test:` Verify whether Γ is an upward drawing.
`Non-Crossing-Test:` Verify whether any two edges of Γ cross.

The `Embedding-Test` can be executed in $O(n)$ time with the techniques described in [5]. Since no geometric test is performed, then the Embedding Test does not affect the overall degree of the checker.

The `Upwardness-Test` can be executed in $O(n)$ time by visiting Γ with a standard visiting procedure. The geometric test involves an immediate comparison of the y-coordinates of the endvertices of the edges. This requires degree 1.

In [5] it is shown that a straight-line undirected drawing whose induced embedding is planar does not have any edge crossings if and only if all faces in the drawing are simple polygons. In the same paper, an efficient algorithms is presented for checking convex planar drawings of undirected graphs. Unfortunately, neither the faces of an upward drawing are in general convex, nor they belong to classes of polygons for which the simplicity test can be easily realized (see also Section 1). Hence, an efficient (linear-time) realization of the Non-Crossing-Test based on the polygon simplicity check for all faces of Γ would imply either using the fairly complex triangulation algorithm by Chazelle [3] or developing an ad-hoc strategy.

We show an optimal degree strategy for the `Non-Crossing-Test`. The strategy can be applied to all upward drawings of digraphs with a regular embedding. As already pointed out in the previous sections, several existing drawing algorithms for upward planar digraphs compute drawings that our checker is able to verify (see Theorem 3). The strategy exploits the relationship between the geometry of Γ and the topological properties of the represented digraph G. Our `Non-Crossing-Test` consists of three steps.

1. We check if Γ is a drawing of a regular upward planar embedded digraph. This can be done by traversing the faces of Γ. During the traversal of a face f: (i) the labeling of f is computed; (ii) the upward-consistency of f is verified; (iii) the regularity of f is verified.
2. We construct the including planar st-digraph of G with its unique (see Theorem 4) complete saturator. Let G' be such including st-digraph. Based on Theorem 5, we use G' to define a total left-to-right ordering in the set of maximal non-intersecting paths of Γ.
3. We explore Γ by visiting one-path-at-time from left to right. To do this, we follow the ordering induced by G'. Namely, we start by visiting the edges of Γ that belong to the leftmost path π_1 of G' from s to t. A new path π of G' is visited (and the edges of Γ that it contains) only after all paths composed by edges that are "to the left" of the edges of π have been already visited. Basically, we follow an "ear-decomposition" of G'. By the theory developed in the previous section, this guarantees that for each edge e' that has been already visited in Γ and for each non-saturating edge e of π, either $e' \overset{l}{\prec}_G e$ or $e' \overset{u}{\prec}_G e$ or $e \overset{u}{\prec}_G e'$.
   We maintain a *rightmost boundary* Π of the drawing. At the first step, Π is the portion of Γ that represents the non-saturating edges of π_1. At the

generic step, the portion of Γ that represents the non-saturating edges of a new path π is compared with Π. If an intersection occurs, then we stop and report the intersection. Else, we compute the new rightmost boundary by *merging* Π and π as follows: for each pair of points p in π and P in Π such that $y(p) = y(P)$, P is deleted from Π and is replaced by p.

We are now ready to analyze the efficiency of our `Non-Crossing-Test`. Regarding Step 1, traversing the faces of Γ can be done in $O(n)$ time. Also, in order to compute the labeling of a face f, for each switch of f it must be checked whether the angle inside f is reflex or not. This can be done with degree 1 by a simple comparison of input coordinates. Regarding Step 2, the construction of G' can be done in $O(n)$ time by exploiting the technique shown in the proof of Lemma 3. Also, computing G' does not require the execution of any geometric tests and thus it does not affect the overall degree of the `Non-Crossing-Test`. Regarding Step 3, the following lemma provides the geometric foundation to analyze its efficiency.

Lemma 4. *Checking whether the non-saturating edges of π intersect the rightmost boundary Π can be done in $O(k)$ time and degree 2. The parameter k is equal to the number of vertices and bends in π plus the number of vertices and bends in Π whose y-coordinates are in the y-interval spanned by π.*

Proof. We follow Π and π from top to bottom with a dove-tail strategy driven by the y-coordinates. At each step a `which-side` test is executed to determine whether a vertex or bend of π is to the right of the corresponding segment of Π. The rightmost boundary Π is represented as follows. A segment r of Π is always a subsegment of an edge e of Γ. Thus, instead of explicitly storing the endpoints of r, we represent r by means of the endpoints of e plus the y-interval spanned by r. This is done to avoid the explicit computation of the x-coordinates of the endpoints of r that would affect the overall degree.

By exploiting such implicit representation of Π, each which-side test corresponds to evaluating the sign of a determinant that defines a multivariate polynomial of degree 2 and such tha all elements of the determinant are either constant values or the coordinates of the vertices and bends of Γ (primitive variables). Since the primitive variables have degree 1 each which-side tests can be executed with degree 2.

The above discussion, Lemma 4, and Theorem 1 imply the following.

Theorem 6. *Let Γ be a polygonal line drawing with n vertices and bends. There exists a checker that verifies whether Γ is an upward planar drawing of a digraph with a regular embedding that runs in $O(n)$ time and has optimal degree 2.*

5 Extensions and Open Problems

Our optimal degree checker can be easily extended to verify quasi-upward planar drawings [1]. Namely, the following theorem can be proved.

Theorem 7. *Let Γ be a polygonal line drawing with n vertices and bends. There exists a checker that verifies whether Γ is a quasi-upward planar drawing of a digraph with a regular embedding that runs in $O(n)$ time and has optimal degree 2.*

Several checking problems remain open in graph drawing. Consider that all graph drawing algorithms guarantee certain geometric properties for the drawings they produce. Such properties are usually called "graphic standards" or "drawing conventions". Some of them appear to be easy to check, while others like checking proximity drawings seem to be much harder. For example, no algorithm is known to efficiently check whether a drawing is a Gabriel drawing [9] or a Relative Neighborhood Drawing [21].

References

[1] P. Bertolazzi, G. D. Battista, and W. Didimo. Quasi upward planarity. Manuscript, 1998.

[2] P. Bertolazzi, G. Di Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 6(12):476–497, 1994.

[3] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[4] B. Chazelle and J. Incerpi. Triangulation and shape-complexity. *ACM Trans. Graph.*, 3(2):135–152, 1984.

[5] O. Devillers, G. Liotta, R. Tamassia, and F. Preparata. Cecking the convexity of polytopes and the planarity of subdivisions. In *Algorithms and Data Structures (Proc. WADS 97)*, volume 1272 of *Lecture Notes Comput. Sci.*, pages 186–199. Springer-Verlag, 1997.

[6] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Comput. Geom. Theory Appl.*, 4:235–282, 1994.

[7] G. Di Battista and R. Tamassia. Algorithms for plane representations of acyclic digraphs. *Theoret. Comput. Sci.*, 61:175–198, 1988.

[8] H. ElGindy, D. Avis, and G. T. Toussaint. Applications of a two-dimensional hidden-line algorithm to other geometric problems. *Computing*, 31:191–202, 1983.

[9] K. R. Gabriel and R. R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[10] M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan. Triangulating a simple polygon. *Inform. Process. Lett.*, 7:175–179, 1978.

[11] F. Harary. *Graph Theory.* Addison-Wesley, Reading, MA, 1972.

[12] S. Hertel and K. Mehlhorn. Fast triangulation of simple polygons. In *Proc. 4th Internat. Conf. Found. Comput. Theory*, volume 158 of *Lecture Notes Comput. Sci.*, pages 207–218. Springer-Verlag, 1983.

[13] D. Kelly. Fundamentals of planar ordered sets. *Discrete Math.*, 63:197–216, 1987.

[14] G. Liotta, F. P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. *SIAM J. Comput.* to appear.

[15] G. Liotta, F. P. Preparata, and R. Tamassia. Robust proximity queries: an illustration of degree-driven algorithm design. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 156–165, 1997.

[16] K. Mehlhorn and S. Näher. *Checking Geometric Structures*, Dec. 1996. Manual.

[17] K. Mehlhorn, S. Näher, T. Schilz, S. Schirra, M. Seel, R. Seidel, and C. Uhrig. Checking geometric programs or verification of geometric structures. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 159–165, 1996.

[18] K. Mehlhorn, S. Näher, T. Schilz, S. Schirra, M. Seel, R. Seidel, and C. Uhrig. Checking geometric programs or verification of geometric structures. Manuscript, 1997.

[19] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, NY, 1985.

[20] G. Toussaint. Efficient triangulation of simple polygons. *Visual Comput.*, 7:280–295, 1991.

[21] G. T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recogn.*, 12:261–268, 1980.