Optimizing Area and Aspect Ratio in Straight-Line Orthogonal Tree Drawings*

Timothy M. Chan†
Dept. of Computer Science
Univ. of Waterloo
tmchan@uwaterloo.ca

Michael T. Goodrich‡
Dept. of Computer Science
Johns Hopkins Univ.
goodrich@cs.jhu.edu

S. Rao Kosaraju§
Dept. of Computer Science
Johns Hopkins Univ.
kosaraju@cs.jhu.edu

Roberto Tamassia¶
Dept. of Computer Science
Brown Univ.
rt@cs.brown.edu

October 30, 2000

Abstract

We investigate the problem of drawing an arbitrary n-node binary tree orthogonally and upwardly in an integer grid using straight-line edges. We show that one can simultaneously achieve good area bounds while also allowing the aspect ratio to be chosen as being a constant or even an arbitrary parameter. In addition, we show that one can also achieve an additional desirable aesthetic criterion, which we call "subtree separation." Our drawings require $O(n \log n)$ area, which is optimal to within a constant factor. An improvement for non-upward drawings is briefly mentioned.

Keywords: graph drawing, binary trees

1 Introduction

Binary trees are, of course, very common structures in many application areas, so obtaining good drawings of binary trees is an important component in a wide variety of visualization tasks. Nevertheless, there are a number of interesting issues regarding binary-tree drawings that are still unresolved, including those related to drawings that optimize the easily-motivated aesthetic criterion of using straight line segments to display edges while also optimizing the area and aspect ratio of the drawing.

Optimizing the area of a drawing is important, because a drawing typically needs to be displayed on a medium of limited area and resolution, such as a terminal window on a workstation screen. Formally, we define the *area* of a drawing to be the area of a smallest rectangle enclosing the drawing. Of course, this assumes a reasonable rule for defining the resolution of a drawing, such as that used in *planar grid* drawings, where all nodes are placed at integer grid points and edges are

*This work is a consequence of the participation of Drs. Goodrich and Tamassia in the 1996 International Workshop on 3D Graph Drawing at Bellairs Research Inst. of McGill University.

†This research was performed while the author was visiting the Center for Geometric Computing at Johns Hopkins University, and it was supported in part by by ARO under grant DAAH04-96-1-0013.

‡This research supported by NSF under Grant CCR-9300079, and by ARO under grant DAAH04-96-1-0013.

§This research supported by NSF under Grant CCR-9508545 and by ARO under grant DAAH04-96-1-0013.

¶This research supported by NSF under Grant CCR-9423847 and by ARO under grant DAAH04-96-1-0013.

drawn as polygonal chains that bend only at integer grid points, without crossing. Additionally, one may wish to restrict the drawing further to be an *orthogonal* drawing, which is a drawing where the polygonal chains representing edges must be composed of only vertical and horizontal segments. When drawn on a rastered device such as a laser printer or computer monitor, such drawings avoid the aliasing effect caused by the "staircased" drawing of edges that are neither vertical nor horizontal.

An optimization parameter that is perhaps equal in importance to area for a drawing, however, is the *aspect ratio* of a drawing's enclosing rectangle, i.e., the ratio of the width to height of the rectangle. A drawing that is, for example, tall and narrow would be difficult to display nicely on a printed page or in a screen window even if the area is reasonably small (although it might fit quite nicely on a cash-register tape). Ideally, the aspect ratio should be a parameter that could be chosen from a large range of values, or, failing that, it should at least be allowed to be that of a "well proportioned" rectangle (e.g., 1, 5/3, 8.5/11, or $(1+\sqrt{5})/2$).

Another aesthetic criterion that may be desirable in some applications is that a tree drawing be *upward*. That is, that the tree be drawn so that no child is placed higher (in the y-direction) than its parent. This criterion is desirable, for example, if the tree represents an inherently hierarchical relationship, such as the organizational structure of a large business.

1.1 Previous related research

There has been a fair amount of research involving area and aspect ratio tradeoffs of tree drawings (e.g., see the annotated bibliography of Di Battista *et al.* [5]). We summarize the previous bounds for planar *polyline* grid drawings, for example, where edges are drawn as polygonal chains that bend only at integer grid points, in Table 1.

Class	Drawing Type	Area	Aspect Ratio(s)	Source
degree-$O(1)$ rooted tree	upward	$\Theta(n)$	$[1/n^{1-\varepsilon}, n^{1-\varepsilon}]$	[6]
binary tree	upward orthogonal	$\Theta(n \log\log n)$	$\log^2 n / n \log\log n$	[6]
degree-4 tree	orthogonal	$\Theta(n)$	1	[8, 14]
degree-4 tree	leaves-on-hull orthogonal	$\Theta(n \log n)$	1	[1]

Table 1: Summary of some area/aspect ratio results for planar polyline grid drawings of trees. (We use ε to denote an arbitrarily small positive constant.)

Notice that each of the area bounds for polyline drawings are tight, to within constant factors, even for upward orthogonal drawings. The related issues for straight-line drawings are not as well-understood, however. We summarize relevant previous results for this class of drawings in Table 2.

We are not aware, for example, of any non-trivial previous work on straight-line orthogonal grid drawings of arbitrary binary trees. This seems to be a fairly serious omission, since straight-line edges are easier for the eye to follow than polyline edges, and orthogonal drawings automatically avoid small angles between edges, which can also cause confusion, and they also avoid aliasing edges drawn on a rasterized device.

Class	Drawing Type	Area	Aspect Ratio(s)	Source
rooted tree	upward layered grid	$O(n^2)$	1	[9]
rooted tree	upward grid	$O(n \log n)$	$\log n/n$ or $n/\log n$	[3, 10]
rooted tree	strictly upward grid	$\Theta(n \log n)$	$\log n/n$	[3]
complete or Fibonacci tree	strictly upward grid	$\Theta(n)$	1	[3, 13]
AVL tree	strictly upward grid	$\Theta(n)$	$[\log^{2+\varepsilon} n/n, n/\log^{2+\varepsilon} n]$	[4]
balanced tree of height $O(\log n)$	upward grid	$O(n \log n)$	$n/\log n$	[3, 10]

Table 2: Summary of previous area/aspect ratio results for planar straight-line grid drawings.

1.2 Subtree separation

There is, in fact, an additional desirable aesthetic property for drawings of binary trees. We say that a region R in the plane is *rectilinearly convex* if the intersection of R and any vertical or horizontal line is connected. For any set S of integer grid points, define the *rectilinear convex hull* of S to be the smallest rectilinearly-convex region containing S. Let $T[v]$ denote the subtree of a tree T that is rooted at node v and contains all the descendents of v in T, i.e., $T[v]$ is the subtree of T induced by v. If, for any disjoint induced subtrees $T[v]$ and $T[w]$ in a binary tree T, the rectilinear convex hulls of $T[v]$ and $T[w]$ are disjoint in a drawing D of T, then we say that D achieves *subtree separation*. This property is desired for binary tree drawings, because it allows the eye to quickly distinguish between different parts of the tree. It also allows for multi-resolutional renderings of a drawing D, so that, for example, if D has too many nodes to all simultaneously fit in a screen window, then D can be rendered up to the resolution of the screen, with some induced subtrees rendered as filled-in rectilinearly-convex regions. Of course, it might not always be possible to achieve subtree separation while also optimizing for other aesthetic criteria. For example, many of the drawings produced by the algorithms of Garg *et al.* [6] do not achieve subtree separation. But it is certainly desirable to achieve this property whenever possible.

1.3 Our results

In this paper we present a general approach, based upon a simple "recursive winding" paradigm, for drawing arbitrary binary trees in small area with good aspect ratio, while satisfying the upward straight-line orthogonal drawing criterion. Intuitively, our recursive winding paradigm draws a binary tree T by laying down a small chain of nodes from left to right until we are near a distinguished node, v_k, and then "winding" by recursively laying out $T[v']$ and $T[v'']$ in the other direction, where v' and v'' denote the children of v_k. We show that the area bound obtained by this approach is optimal to within a constant factor. Specifically, we establish the following results:

- Every n-node binary tree has a planar upward orthogonal straight-line grid drawing D with $O(n \log n)$ area for any aspect ratio in the range $[\log n/n, n/\log n]$. Moreover, D can be made to achieve subtree separation, and it can be produced in $O(n)$ time.
- There are n-node binary trees that require $\Omega(n \log n)$ area in any planar upward orthogonal straight-line drawing that achieves an aspect ratio in the range $[1/n^{1-\varepsilon}, n/\log n]$, for any fixed $\varepsilon > 0$.

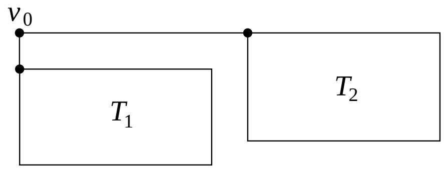


Figure 1: Drawing of a binary tree with $O(\log n)$ height and $O(n)$ width.

We also comment on how the area bound can be improved if we drop the upward condition. Interestingly, our recursive winding approach can be used to prove the existence of a (non-upward) orthogonal straight-line drawing of T with $O(n \log\log n)$ area. This particular result was independently obtained by Shin *et al.* [11], who went on to achieve arbitrary aspect ratio for these non-upward drawings. Their strategy is quite similar to ours but has a somewhat more complicated description. (In fact, subsequent papers by Shin *et al.* [7, 12] have adopted our framework to study related problems.)

Remark: The original proof of the $O(n \log n)$ area result in the extended abstract of this paper [2] was flawed, as pointed to us by T. Shermer (personal communication, July 2000). We have modified our upward algorithm in the present version to correct this error and, at the same time, extend the range of aspect ratios from constant to arbitrary.

2 Upward Drawings with Arbitrary Aspect Ratios

In this section, we present our recursive winding paradigm and obtain an algorithm that can produce upward straight-line orthogonal tree drawings of $O(n \log n)$ area with any feasible aspect ratio the user desires:

Theorem 1: *Given any binary tree T with n nodes and a parameter $2 \leq A \leq n$, there is a planar upward straight-line orthogonal grid drawing of T with $O((n/A)\log A)$ height and $O(A \log n/ \log A)$ width. Such a drawing can be constructed in $O(n)$ time, and it achieves subtree separation.*

Proof: Without loss of generality, assume that each internal node has degree 2. Given an internal node v, let *left*(v) and *right*(v) denote the left child and the right child of v respectively. Let $T[v]$ again denote the subtree of T rooted at v, and let $N[v]$ be the number of leaves in $T[v]$. Arrange the tree so that $N[\mathit{left}(v)] \leq N[\mathit{right}(v)]$ at every node v. This preprocessing requires only linear time. We first review the following lemma (all logarithms in this paper have base 2):

Lemma 2 [3, 10]: *If T has n leaves, then there is a planar upward orthogonal straight-line grid drawing of T with height at most $\lfloor \log n \rfloor$ and width $n-1$. The root is placed at the upper left-hand corner and the construction time is $O(n)$.*

Proof: The construction is recursive. If $n = 1$, the drawing is trivial. Suppose $n > 1$ and v_0 is the root of T. Letting $T_1 = T[\mathit{left}(v_0)]$ and $T_2 = T[\mathit{right}(v_0)]$, we can draw T as shown in Figure 1, where the subtrees T_1 and T_2 are drawn recursively. Since $N[\mathit{left}(v_0)] \leq N[\mathit{right}(v_0)]$, an induction argument shows that the height of the drawing is bounded by $\log n$. $\square$

Next we analyze a recurrence relation.

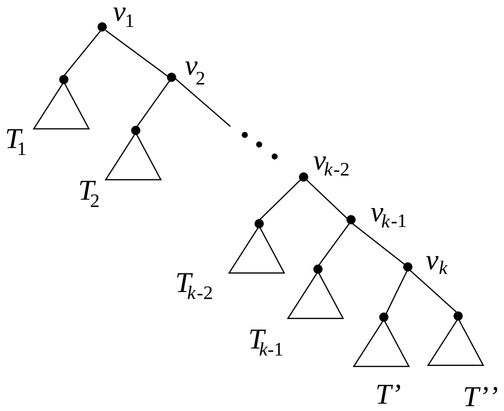


Figure 2: The binary tree T.

Lemma 3: *Suppose $A > 1$ and f is a function such that*

- *if $n \le A$, then $f(n) \le 1$; and*
- *if $n > A$, then $f(n) \le f(n') + f(n'') + 1$ for some $n', n'' \le n - A$ with $n' + n'' \le n$.*

Then $f(n) < 4n/A - 1$ for all $n > A$.

Proof: The proof is by induction. Suppose the lemma is true for n' and n''. If both $n', n'' \le A$, then $f(n) \le 3 < 4n/A - 1$. If $n' \le A$ and $n'' > A$, then

$$f(n) \le f(n'') + 2 < 4n''/A + 1 \le 4(n - A)/A + 1 < 4n/A - 1.$$

Finally, if both $n', n'' > A$, then

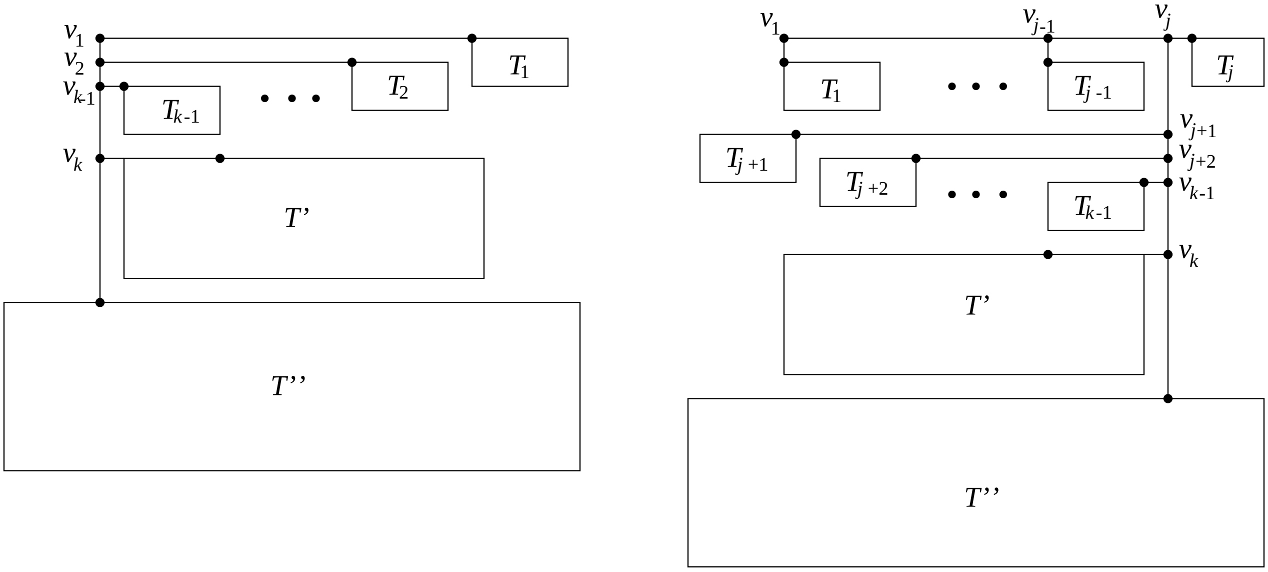
$$f(n) \le f(n') + f(n'') + 1 < 4n'/A + 4n''/A - 1 \le 4n/A - 1. \qquad \square$$

Let $1 \le B \le A$ be a fixed parameter to be determined later. We now prove Theorem 1 by a recursive algorithm. The drawing produced will obey the additional condition that the root is placed B units to the right of the upper left-hand corner of an enclosing rectangle, and no nodes come between the root and this upper left-hand corner. Let $H(n)$, $W(n)$, and $T(n)$ denote the height and width of the enclosing rectangle for the drawing of a tree with n leaves. If $n \le A$, then we use the scheme in Lemma 2 (and stretch the enclosing rectangle by B units to the left to satisfy the above condition). This provides the base case:

$$H(n) \le \log A \text{ and } W(n) \le A + B \quad \text{if } n \le A.$$

Suppose $n > A$. Define a sequence $\{v_i\}$ of nodes as follows: v_1 is the root and $v_{i+1} = \mathit{right}(v_i)$ for $i = 1, 2 \ldots$ Since $N[v_1], N[v_2], \ldots$ is a strictly decreasing sequence of integers, there is an index k with $N[v_k] > n - A$ and $N[v_{k+1}] \le n - A$. Let $T_i = T[\mathit{left}(v_i)]$ and $n_i = N[\mathit{left}(v_i)]$ for $i = 1, \ldots, k-1$. Let $T' = T[\mathit{left}(v_k)]$, $T'' = T[\mathit{right}(v_k)]$, $n' = N[\mathit{left}(v_k)]$, and $n'' = N[\mathit{right}(v_k)]$. (See Figure 2.) Note that $n' \le n''$, since T is "right-heavy." The following properties then hold:

1. $n_1 + \cdots + n_{k-1} = n - N[v_k] < A$,



(a) $k \leq A/B + 2$ (b) $k > A/B + 2$

Figure 3: Upward drawing of T with $O((n/A)\log A)$ height and $O(A \log n / \log A)$ width.

2. $\max\{n', n''\} = N[v_{k+1}] \leq n - A$, and
3. $n' \leq n/2$.

Now, consider the planar upward orthogonal straight-line grid drawing of T in Figure 3(a) or (b), depending on whether $k \leq A/B+2$ or $k > A/B+2$. As required, the root v_1 is placed B units to the right of the upper left-hand corner (by stretching if necessary). The subtrees $T_1, \ldots, T_{k-1}$ are all drawn according to Lemma 2, while the subtrees T' and T'' are drawn recursively.

Case (a): $k \leq A/B + 2$. The drawing has the following height and width:

$$\begin{aligned} H(n) &\leq H(n') + H(n'') + \log A + k + 1 \\ W(n) &\leq \max\{n_1 + \cdots + n_{k-1} + B,\ W(n') + B + 1,\ W(n'')\}. \end{aligned}$$

Case (b): $k > A/B + 2$. Here, the drawings of $T_{j+1}, \ldots, T_{k-1}$, T', and T'' are all reflected; for example, the root of the subtree T'' is now placed B units left of the right side of its enclosing rectangle (this is the "recursive winding"). We choose an index $j \in \{k - \lceil A/B \rceil, \ldots, k-1\}$ such that $n_j \leq B$ (so that the right side of the overall enclosing rectangle is indeed determined by T''). By property 1, such an index exists (otherwise, $n_1 + \cdots + n_{k-1}$ would exceed $\lceil A/B \rceil B \geq A$, a contradiction). Therefore, the drawing can be made with the following height and width bounds:

$$\begin{aligned} H(n) &\leq H(n') + H(n'') + 2\log A + k - j + 3 \\ W(n) &\leq \max\{n_1 + \cdots + n_{j-1} + 2B,\ n_{j+1} + \cdots + n_{k-1} + B,\ W(n') + B + 1,\ W(n'')\}. \end{aligned}$$

In any case, by properties 1 and 3, we can write the recurrences as

$$\begin{aligned} H(n) &\leq H(n') + H(n'') + O(\log A + A/B) \\ W(n) &\leq \max\{O(A + B),\ W(n/2) + O(B),\ W(n'')\}. \end{aligned}$$

We can see that $W(n) = O(B \log n + A)$. By property 2 and an application of Lemma 3, we can also conclude that $H(n) = O((n/A)(\log A + A/B))$. Setting $B = \lceil A/\log A \rceil$ proves the theorem.

The construction time is clearly linear in the number of nodes (since we spend constant time per node). Moreover, by induction, the drawing satisfies subtree separation. □

3 A Lower Bound for Upward Drawings

In this section, we show that the area bound of the preceding algorithm is the best possible by exhibiting binary trees that require $\Omega(n \log n)$ area for any planar upward straight-line orthogonal drawing with good aspect ratio. Specifically, we prove the theorem below. (If the aspect ratio exceeds $1/n^{1-\varepsilon}$, then the width must be at least $n^{\varepsilon/2}$, and we can, for instance, set $A = \lceil n^{\varepsilon/4} \rceil$ to obtain an $\Omega((1/\varepsilon) n \log n)$ area bound.)

Theorem 4: *Given any n and a parameter $2 \le A \le n$, there exists a binary tree T with n nodes, such that any planar straight-line orthogonal upward grid drawing D with width $W \ge A \log A$ has height $H = \Omega((n/W) \log A)$.*

Proof: We begin with a simple lemma that gives a logarithmic lower bound on the height of the drawing of a complete tree. Here, a *rightward (resp. leftward)* path refers to a path that is drawn so that the x-coordinates of its nodes are non-decreasing (resp. non-increasing).

Lemma 5: *If T is an n-node complete binary tree, then any planar upward straight-line orthogonal grid drawing D of T contains both a rightward and a leftward root-to-leaf path of height at least $\lfloor (\log n)/2 \rfloor$.*

Proof: The proof is based upon an induction argument similar to that used by Crescenzi *et al.* [3] for upward non-orthogonal grid drawings. Consider the two children v_1 and v_2 of the root v_0. If one of them, say v_1, is placed directly below v_0 in D, then we can generate a rightward path from v_1 recursively and append v_0 in front to increase the path height by one. On the other hand, if both v_1 and v_2 are placed on the same level as v_0, with v_1 to the right, then one of the children of v_1, say v_{11}, must be placed directly below v_1, and we can generate a rightward path from v_{11} recursively and append v_0 and v_1 in front to increase the path height by one. By induction, this implies the existence of a rightward path from the root of length $\lfloor (\log n)/2 \rfloor$. The other part of the lemma is symmetric. □

The proof of Theorem 4 is based upon a non-trivial adaptation of a "chain pinning" argument of Garg *et al.* [6], originally used to establish an $\Omega(n \log \log n)$ area bound on upward polyline orthogonal drawings (here, we basically use a different parameter, and by exploiting the straight-line condition with the above lemma, we also simplify the proof somewhat). We choose the tree T defined by a chain C of $n/2$ nodes $v_1, v_2, \ldots, v_{n/2}$, where every node v_i with i being a multiple of A—called a *joint*—is attached a complete binary tree with A nodes (rooted at a child of the joint).

In the drawing D of this tree T, the chain C is drawn as a sequence of horizontal segments mixed with vertical drops. Call a joint a *boundary joint* if it is the leftmost or rightmost joint in its row; otherwise, call it an *interior joint*. Clearly, there are $O(H)$ boundary joints. We claim that any rectangle R of height $\lfloor(\log A)/2\rfloor$ and width $A-1$ contains at most one interior joint; therefore, the number of interior joints is $O(HW/(A\log A))$. Since there are $\Theta(n/A)$ joints in total,

$$\frac{n}{A} \;=\; O\left(H+\frac{HW}{A\log A}\right) \;=\; O\left(\frac{HW}{A\log A}\right),$$

and the theorem follows.

To prove the claim, suppose the rectangle R contains two interior joints v_i and v_j. Since R has width $A-1$, each row inside R has at most one joint. Now, in the drawing of C, the horizontal segment S_i through v_i must completely cross R (otherwise, v_i would be a boundary joint), and similarly, the horizontal segment S_j through v_j must completely cross R. Without loss of generality, say S_i is above S_j. Their distance is upper-bounded by the height of R, $\lfloor(\log A)/2\rfloor$. By Lemma 5, the subtree attached to v_i has both a leftward and rightward path of height $\lfloor(\log A)/2\rfloor$, so the chain C (in particular, the segment S_j) cannot reach the nodes at distance $\leq \lfloor(\log A)/2\rfloor$ directly below v_i: a contradiction. □

4 Non-Upward Drawings

In the non-upward case, the area bound can actually be improved by our recursive winding paradigm with a slightly simpler construction. In the following, ignoring the aspect ratio, we can, for example, take $A=\lceil\log n\rceil$ and obtain a drawing with width $O(\log n)$ and height $O((n/\log n)\log\log n)$, and hence, area $O(n\log\log n)$.

Theorem 6: *Given any binary tree T with n nodes and a parameter $2\leq A\leq n$, there is a planar straight-line orthogonal grid drawing of T with $O((n/A)\log A)$ height and $O(\log n+A)$ width. Such a drawing can be constructed in $O(n)$ time, and it achieves subtree separation.*

Proof: We use the same notation as in Section 2. This time, the drawing will satisfy the condition that the root is placed on the left side of the enclosing rectangle, and no nodes come between the root and the upper left-hand corner. The drawing of T is depicted in Figure 4(a), (b), or (c), depending on whether $k=1$, $k=2$, or $k>2$. The subtrees $T_1,\ldots,T_{k-1}$ are drawn according to Lemma 2, where in *Case* (c), the drawing of T_{k-1} is rotated 180 degrees. As in the previous construction, the subtrees T' and T'' are drawn recursively, and in *Case* (c), they are reflected, so that their roots are placed on the right side of their respective rectangles.

This all implies that this (non-upward) planar straight-line orthogonal grid drawing can be made with the following bounds on the height and width:

$$\begin{aligned} H(n) &\leq H(n')+H(n'')+2\log A+4 \\ W(n) &\leq \max\{n_1+\cdots+n_{k-1},\; W(n')+1,\; W(n'')\}. \end{aligned}$$

By properties 1 and 3, we can rewrite the recurrences as

$$\begin{aligned} H(n) &\leq H(n')+H(n'')+O(\log A) \\ W(n) &\leq \max\{A,\; W(n/2)+1,\; W(n'')\}. \end{aligned}$$

It follows that $W(n)=O(\log n+A)$, and again by property 2 and Lemma 3, $H(n)=O((n/A)\log A)$. The construction time is linear and the drawing satisfies subtree separation. □

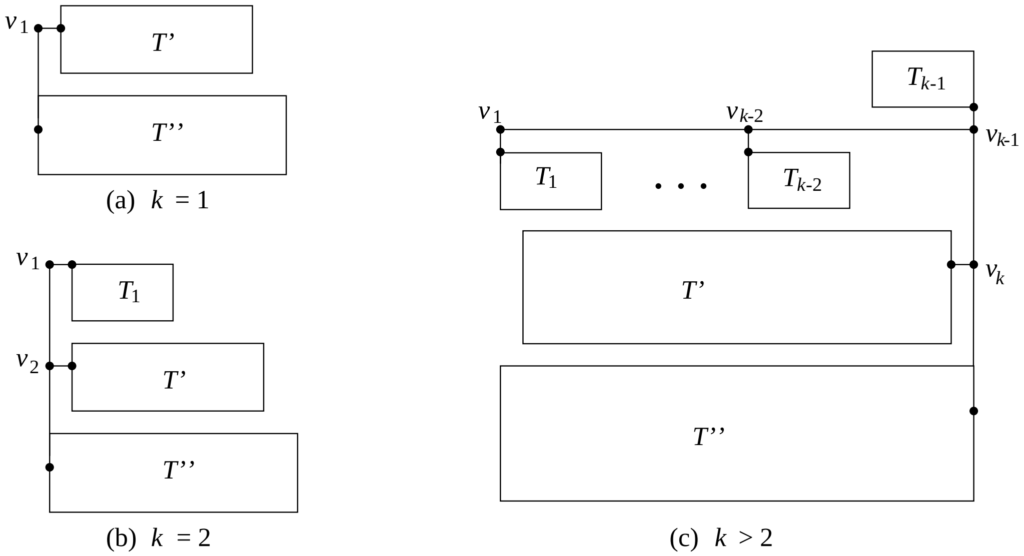


Figure 4: Non-upward drawing of T with $O((n/A)\log A)$ height and $O(\log n + A)$ width.

5 Conclusion

We have investigated several issues related to space-efficient planar straight-line orthogonal grid drawings of arbitrary binary trees. In the case of upward drawings we have established a tight upper and lower bounds of $\Theta(n \log n)$ on the area needed, with arbitrary aspect ratios. A modification of the same idea in fact yields a non-upward drawing with area $O(n \log\log n)$, as independently proved by Shin *et al.* [11]. Some interesting problems that remain open include the following:

- Can one prove that there are binary trees requiring $\Omega(n \log n)$ area for any planar upward straight-line orthogonal grid drawing regardless of the aspect ratio? (Currently, the best lower bound is still the $\Omega(n \log\log n)$ bound for upward polyline orthogonal drawings [6].)
- Are there binary trees that require $\Omega(n \log\log n)$ area for any (non-upward) planar straight-line orthogonal grid drawing?

Acknowledgements

We would like to thank Stina Bridgeman, Marek Chrobak, and Sue Whitesides for several stimulating discussions and e-mail exchanges relating to the topics of this paper. We also thank Tom Shermer for pointing out the error in the preliminary version of this paper.

References

[1] R. P. Brent and H. T. Kung. On the area of binary tree layouts. *Inform. Process. Lett.*, 11:521–534, 1980.

[2] T. M. Chan, M. T. Goodrich, S. R. Kosaraju, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. In S. North, editor, *Graph Drawing (Proc. GD '96)*, volume 1190 of *Lecture Notes Comput. Sci.*, pages 63–75. Springer-Verlag, 1997.

[3] P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Comput. Geom. Theory Appl.*, 2:187–200, 1992.

[4] P. Crescenzi, P. Penna, and A. Piperno. Linear-area upward drawings of AVL trees. *Comput. Geom. Theory Appl.*, 9:25–42, 1998. (special issue on Graph Drawing, edited by G. Di Battista and R. Tamassia).

[5] 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.

[6] A. Garg, M. T. Goodrich, and R. Tamassia. Planar upward tree drawings with optimal area. *Internat. J. Comput. Geom. Appl.*, 6:333–356, 1996.

[7] S. K. Kim. Logarithmic width, linear area upward drawing of AVL trees. *Inf. Proc. Lett.*, 63:303–307, 1997.

[8] C. E. Leiserson. *Area-efficient graph layouts (for VLSI).* ACM Doctoral Dissertation Award Series. MIT Press, Cambridge, MA, 1983.

[9] E. Reingold and J. Tilford. Tidier drawing of trees. *IEEE Trans. Softw. Eng.*, SE-7(2):223–228, 1981.

[10] Y. Shiloach. *Arrangements of Planar Graphs on the Planar Lattice.* PhD thesis, Weizmann Institute of Science, 1976.

[11] C.-S. Shin, S. K. Kim, and K.-Y. Chwa. Area-efficient algorithms for straight-line tree drawings. *Comput. Geom. Theory Appl.*, 15:175–202, 2000.

[12] C.-S. Shin, S. K. Kim, S.-H. Kim, and K.-Y. Chwa. Algorithms for drawing binary trees in the plane. *Inf. Proc. Lett.*, 66:133–139, 1998.

[13] L. Trevisan. A note on minimum-area drawing of complete and Fibonacci trees. *Inf. Proc. Lett.*, 57:231–236, 1996.

[14] L. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Comput.*, C-30(2):135–140, 1981.