Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 17, no. 2, pp. 103–128 (2013)
DOI: 10.7155/jgaa.00287

Drawing Unordered Trees on k-Grids

Christian Bachmaier[1] *Marco Matzeder*[1]

[1]University of Passau, 94030 Passau, Germany

Abstract

We present almost linear area bounds for drawing trees on the octagonal grid. For complete 7-ary trees we establish an upper and lower bound of $\Theta(n^{1.129})$ and for complete ternary trees the bounds of $\mathcal{O}(n^{1.048})$ and $\Theta(n)$, where the latter needs edge bends. For arbitrary ternary trees we obtain an upper bound of $\mathcal{O}(n \log \log n)$ with bends and good aspect ratio by applying the recursive winding technique. We explore the unit edge length and area complexity of drawing unordered trees on k-grids with $k \in \{4, 6, 8\}$ and generalize the $\mathcal{NP}$-hardness results of the orthogonal grid to the octagonal and hexagonal grids.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
April 2012	July 2012	August 2012	December 2012	December 2012

Published: January 2013

Article type: Regular paper — Communicated by: S.-i. Nakano

A preliminary version [2] was presented at the Workshop on Algorithms and Computation, WALCOM 2012.

E-mail addresses: bachmaier@fim.uni-passau.de (Christian Bachmaier) matzeder@fim.uni-passau.de (Marco Matzeder)

1 Introduction

Trees are a fundamental data structure in computer science to represent hierarchies. Amongst others they are used as family trees in social networks or inheritance structures in UML-diagrams. Their visualization is an important field in graph drawing [8, 9, 11, 15, 16, 24]. Often trees are unordered, e. g., flow charts. Then, it is not necessary that a drawing reflects a given child order. For readable and comprehensible drawings in traditional hierarchical style the following aesthetics are established [19, 23, 25]: y-coordinates of the vertices correspond to their depth, centered parents over their children, minimal distance between vertices, integral coordinates, no (too) small angles between edges incident on a common vertex, maintaining the order, planarity, and identically drawn isomorphic subtrees up to reflection. These criteria exclude recursive winding techniques as they were studied by Chan et al. [8, 9]. Marriott and Stuckey [21] have shown that for unordered binary trees it is $\mathcal{NP}$-hard to determine a hierarchical drawing with minimal width. The same was shown by Supowit and Reingold [25] for order-preserving drawings. The common drawing algorithm for binary trees was introduced by Reingold and Tilford [23] and generalized to d-ary trees by Walker [4, 7, 27], which all satisfy the above aesthetic criteria.

The hierarchical drawing methods enforce placing the vertices at grid points. All these approaches allow drawing trees of high degree, such that the angles between incident edges may be very small. Restricting the degree of trees allows to draw along a finite set of directions, e. g., four directions on the orthogonal grid. This grid was widely investigated in literature [3, 8–10, 13, 15, 16, 24, 26]. The 6- and the 8-grid with additional axes are used to draw trees with higher degree [1, 6, 18]. A motivation for such grids are discrete representations of radial drawings [11]. In our companion paper [6] we showed that it is $\mathcal{NP}$-hard to determine the existence of an order-preserving tree drawing within a given area on the k-grid with $k \in \{4, 6, 8\}$. Now we translate the $\mathcal{NP}$-hardness to the unordered case. Bhatt and Cosmadakis [3] showed that it is $\mathcal{NP}$-hard to determine if a tree of degree up to four has an unit edge length drawing on the orthogonal grid. This result can also be proven by the logic engine approach [12]. For binary trees this was shown by Gregori [17]. $\mathcal{NP}$-hardness results for minimum area were presented in [5, 20]. We already have claimed an equivalent result on the 6-grid for trees with degree up to six [1], however, in contrast to here without a formal proof. In the plane only two grid axes are linear independent. This shall cause some problems for compacting drawings on higher order grids containing more than 2 axes which do not occur on the 4-grid. Furthermore, the degree of difficulty increases with the number of additionally available discrete directions.

The remainder is organized as follows. After some definitions in Sect. 2 we present a tight area bound for drawing complete 7-ary trees in Sect. 3. Afterwards, we show an almost linear upper area bound for straight-line drawings and bounds for drawings of ternary trees with bends on the 8-grid in Sect. 4. Table 1 summarizes known area bounds for unordered complete trees on k-grids.

Table 1: Area bounds for complete d-ary trees on k-grid

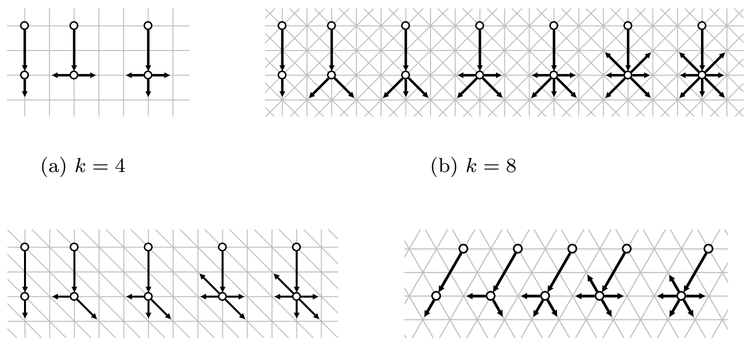
	$k = 4$	$k = 6$	$k = 8$
$d = 2$	$\Theta(n)$ [10]	$\Theta(n)$ [10]	$\Theta(n)$ [10]
$d = 3$	$\mathcal{O}(n^{1.262})$ [13]	$\mathcal{O}(n^{1.262})$ [1]	$\mathcal{O}(n^{1.048})$, Theorem 2 $\mathcal{O}(n)$, with bends, Theorem 3
$d = 4$	—	$\mathcal{O}(n^{1.585})$ [1]	$\mathcal{O}(n^{1.585})$ [1]
$d = 5$	—	$\mathcal{O}(n^{1.366})$ [1]	$\mathcal{O}(n^{1.366})$ [1]
$d = 6$	—	—	$\mathcal{O}(n^{1.227})$ [22]
$d = 7$	—	—	$\Theta(n^{1.130})$, Theorem 1

Finally, we show in Sect. 5 that it is $\mathcal{NP}$-hard to decide whether or not there is a unit edge length drawing for arbitrary trees with degree $k \in \{6, 8\}$ on the k-grid and whether or not there is a drawing within a given area.

2 Preliminaries

The *orthogonal* or *4-grid* is the infinite planar undirected graph $G = (V, E)$ whose vertices V have integral coordinates and whose edges E link vertex pairs with vertical or horizontal unit distance. Throughout the paper we use a Cartesian coordinate system with ascending y-coordinates downwards. We extend the 4-grid with its four directions to the *hexagonal* or *6-grid* [1, 6, 18] with six directions by adding an edge $\{u, v\}$ for each $u \in V$ on coordinates (x, y) and $v \in V$ on $(x + 1, y + 1)$. The *octagonal* or *8-grid* is a 6-grid with additional edges $\{u, v\}$ between each $u \in V$ on (x, y) and $v \in V$ on $(x + 1, y - 1)$. We call these grids *k-grids* with $k \in \{4, 6, 8\}$. The *distance* between vertices $u, v \in V$ with coordinates (u_x, u_y) and (v_x, v_y) on a k-grid is defined by the maximum metric $d(u, v) = \max(|u_x - v_x|, |u_y - v_y|)$. A *path* $(v_1, \ldots, v_n)$ is a sequence of vertices with edges (v_i, v_{i+1}) and $i \in \{1, \ldots, n - 1\}$. A path is *straight* if the edges have the same direction. Let $T = (V, E)$ be a (rooted) *unordered tree*. If each vertex $v \in V$ has an outdegree of at most d, we call T a *d-ary tree*. An *embedding* $\Gamma(T)$ of a d-ary tree $T = (V, E)$ on a k-grid with $d < k$ is a mapping Γ which specifies distinct integer coordinates $\Gamma(v) = (x, y)$ for each vertex $v \in V$. Γ maps an edge $e \in E$ onto a (straight) path of grid edges $\Gamma(e)$ between its endpoints. The *length* of an edge $e \in E$ is the distance between its incident vertices and the *length* of a path is the sum of its edge lengths. We use the terms *drawing* and embedding synonymously. The *area* on a k-grid is the size of the smallest bounding rectangle and the *aspect ratio* is the quotient of its height and its width.

The following definitions of drawing styles are in accordance to [6], where we replace "O" for ordered by "U" for unordered. An *U_k-drawing* is a drawing of an unordered d-ary tree on a k-grid with $d < k$. A tree drawing is *locally uniform* if



(a) $k = 4$ (b) $k = 8$

(c) $k = 6$ (asymmetric) (d) $k = 6$ (symmetric)

Figure 1: Patterns on the k-grids

for each vertex its outgoing edges have identical length. We call a locally uniform U_k-drawing UL_k-*drawing*. In a *pattern drawing* of a d-ary tree on the k-grid, the outgoing edges of each vertex are axially symmetric with respect to the incoming edge. All patterns are categorized by their outdegree. They are listed in Fig. 1 for the various k-grids. An U_k-drawing using patterns is called UP_k-*drawing*. Combining these properties we obtain locally uniform pattern drawings, called ULP_k-*drawings*. Here the children of a vertex are positioned symmetrically, which corresponds to placing the parent centered over its children.

3 Complete 7-ary Trees

In this section we investigate drawing complete 7-ary trees on the 8-grid. Similar to the results of [1], we establish an upper and lower bound for the area needed for complete trees.

Theorem 1 *The upper and the lower bound for the area of drawings of complete 7-ary trees with* n *vertices on the 8-grid is* $\Theta(n^{\log_7 9})$.

Proof: We construct the drawing of the tree recursively. In the initial case the tree has height $h = 0$. In the construction step $h \to h + 1$ the side length (in grid points) of the planar drawing grows by a factor of three, see Fig. 2. Thus, the area is in $\mathcal{O}(9^h)$. Since $h = \log_7 n$, the area is $\mathcal{O}(9^{\log_7 n}) \subset \mathcal{O}(n^{1.129})$.

Let $\Gamma(T(h))$ be an U_k-drawing of a complete 7-ary tree of height h with root r on the 8-grid. W. l. o. g. we assume that r is placed at the origin. We proof by induction on h that at least seven of the four corner extreme points $(\pm\frac{3^h-1}{2}, \pm\frac{3^h-1}{2})$ and four center extreme points $(\pm\frac{3^h-1}{2}, 0)$ and $(0, \pm\frac{3^h-1}{2})$ are occupied by a vertex or an edge of $T(h)$. Note that one of the corner extreme points may be used for the incoming edge of the root of $T(h)$. Clearly, the

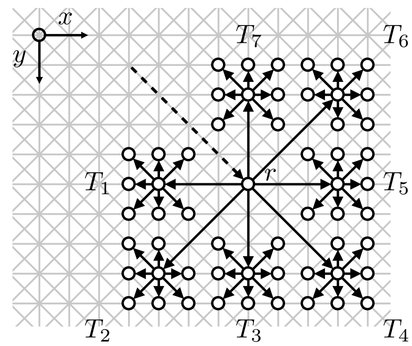


Figure 2: Complete 7-ary tree

statement holds for the induction bases $h = 0$ and $h = 1$. Let $T_1, \dots, T_7$ be the seven (complete) subtrees of r with height $h-1$ and roots $r_1, \dots, r_7$. W. l. o. g. let the outgoing edges of r $(r, r_1), \dots, (r, r_7)$ point to any of the eight possible edge directions except to the north-west. This allows us to assume that the numbering of the T_is is counter-clockwise starting with the west.

Assume for contradiction that the grid point $p = (\frac{3^h-1}{2}, 0)$ is not occupied by $\Gamma(T(h))$. Hence, the subtree T_5 with incoming edge (r, r_5) pointing to the east does not occupy p. By induction, the side lengths of the drawing of T_5 are at least $3^{h-1} - 1$. Then, one corner extreme point of T_5 overlaps the diagonal axis from r to the north-east. Therefore, it is not possible to use this diagonal direction for T_6. By symmetric arguments, the same holds for T_1, T_3, and T_7.

It remains to show that the corner extreme points are occupied. Assume for contradiction that the grid point $q = (\frac{3^h-1}{2}, \frac{3^h-1}{2})$ is not occupied by $\Gamma(T(h))$. Hence, the subtree T_4 with incoming edge (r, r_4) pointing to the south-east does not occupy q. As a consequence, its extreme points are placed at least one unit to the north and/or to the west. W. l. o. g. we assume that it is displaced one unit to the north (both other directions are symmetric). As the side lengths of T_4 and T_5 are at least $3^{h-1} - 1$ and T_4 and T_5 do not overlap, T_5 overlaps the diagonal axis from r to the north-east. Therefore, it is not possible to use this diagonal direction for T_6. The same can be shown for T_2 and T_6 by symmetric arguments. □

Corollary 1 *There is a linear time algorithm to draw a complete 7-ary rooted tree with n vertices on the 8-grid in $\mathcal{O}(n^{1.129})$ area and with aspect ratio 1.*

4 Ternary Trees

First we consider complete ternary trees and afterwards arbitrary ternary trees on the 8-grid.

4.1 Complete Ternary Trees

Each complete ternary tree can be drawn on the 4-grid within $\mathcal{O}(n^{1.262})$ area [13]. For strictly upward drawings on the 6-grid there is a tight bound of $\Theta(n^{1.262})$ [1]. In *upward drawings* the edges point piecewise monotonically in a common direction. We present an almost linear upper bound for complete ternary trees on the 8-grid using all 8 directions.

Theorem 2 *There is a linear time algorithm to draw a complete ternary tree with n vertices on the 8-grid in $\mathcal{O}(n^{1.048})$ area and with aspect ratio 1.*

Proof: We construct the tree $T(h)$ with height h recursively. Figure 3 shows one recursion step for $i \to i+4$ with $i \le h$. Initially, for $i = h \bmod 4 \in \{0, \dots, 3\}$, we draw the tree $T(i)$ within a square of constant area with the root at a corner. In a recursion step $i \to i+4$ there are 81 complete trees with height i which we draw within a square with side length $S(i)$. Let $c = 8$ be the number of additionally inserted columns (rows) which are used for wiring, i. e., connecting the subtrees with their parents. Then, the side length is $S(i+4) = 10 \cdot S(i) + c \le 10^{\lceil i/4 \rceil} + (c \cdot \sum_{i=0}^{\lceil i/4 \rceil} 10^i) < 10^{\lceil i/4 \rceil} + c \cdot 10^{\lceil i/4 \rceil + 1}$. Thus, $S(h) \in \mathcal{O}(10^{h/4})$ and the area of $T(h)$ is in $\mathcal{O}(100^{h/4})$. The height of a complete ternary tree is $h = \log_3 n$. Therefore, the needed area is in $\mathcal{O}(100^{(\log_3 n)/4}) = \mathcal{O}(n^{(\log_3 100)/4}) \subset \mathcal{O}(n^{1.048})$. $\square$

Theorem 3 *There is a linear time algorithm to draw a complete ternary tree with at most one bend per edge on the 8-grid and the 6-grid within $\Theta(n)$-area. The drawing is strictly upward, has constant aspect ratio, and less than $\frac{n}{9}$ bends.*

Proof: We inductively define a drawing of a complete ternary tree $T(h)$ with height h. For trees with height $h = 0$ and $h = 1$ the drawing is shown in Fig. 4. In the construction step $h \to h+1$ we compose the complete subtrees $T_1(h)$, $T_2(h)$, and $T_3(h)$ rooted at r_1, r_2, and r_3 to a tree $T(h+1)$ with root r and connecting edges $(r, r_1), (r, r_2)$, and (r, r_3). This is done either vertically (Fig. 5(a)) or horizontally (Fig. 5(b)), alternating with odd and even h. Note that $T_1(h), T_2(h), T_3(h)$ are drawn identically and, thus, their drawings have identical dimensions. We denote the width and the height (number of occupied grid points) of the drawing of a tree with height h with $W(h)$ and $H(h)$, respectively.

We place r at the origin. If h is odd, we apply the vertical construction and place r_1 at $(0, 2H(h)+2)$, r_2 at $(1, H(h)+1)$, and r_3 at $(0, 2)$. Otherwise, if h is even, we apply the horizontal construction and place r_1 at $(0, 2)$, r_2 at $(W(h)+1, 1)$, and r_3 at $(2W(h)+2, 0)$. For an example see Fig. 4.

In the vertical step the height $H(h+1) = 3H(h)$ and the width $W(h+1) = W(h)+2$. For the horizontal step symmetrically $H(h+1) = H(h)+2$ and $W(h+1) = 3W(h)$. The correctness of these recursions can be shown by induction. After eliminating the recursion we substitute h by $\log_3(2n+1) - 1$, since $n = \frac{3^{h+1}-1}{2}$ in 3-ary trees, and obtain

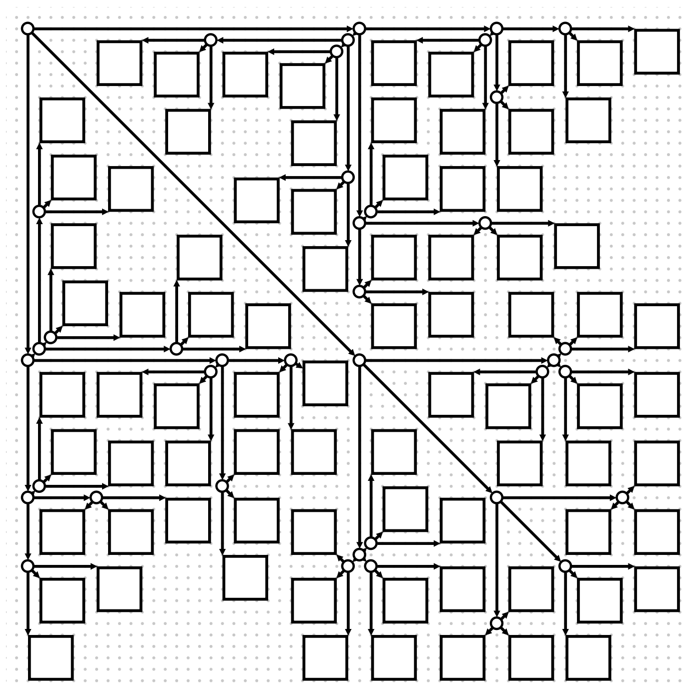


Figure 3: Scheme to draw complete ternary trees on 8-grids recursively (grid lines omitted)

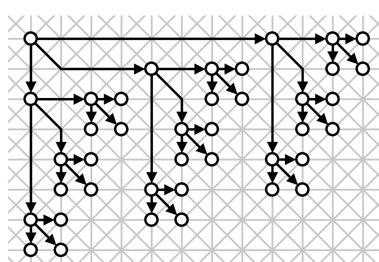
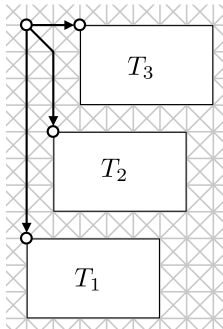
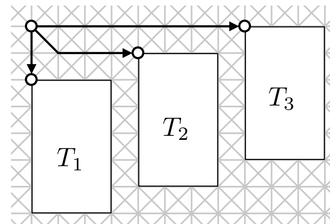


Figure 4: Complete 3-ary tree with bends



(a) Vertical



(b) Horizontal

Figure 5: Construction steps for Theorem 3

$$H(h) = 2 \cdot 3^{\lfloor \frac{h}{2} \rfloor + 1} - 2 - (-1)^h \in \mathcal{O}(\sqrt{n}), \quad (1)$$
$$W(h) = 5 \cdot 3^{\lfloor \frac{h-1}{2} \rfloor} + 2 + (-1)^h \in \mathcal{O}(\sqrt{n}). \quad (2)$$

Finally, we show that the overall number of edge bends b is smaller than $\frac{n}{9}$. In the drawing of a tree $T(h)$ with $n = \frac{3^{h+1}-1}{2}$ vertices $b = \frac{3^{h-1}-1}{2}$ as each recursion step produces exactly one bend. Solving these equations results in $b = \frac{n-4}{9}$. □

4.2 Arbitrary Ternary Trees

We transfer the *recursive winding* approach of Chan et al. [8, 9] for drawing binary trees on the 4-grid to ternary trees on the 8-grid introducing bends. On the 4-grid the algorithm constructs straight-line upward planar drawings respecting subtree separation with $\mathcal{O}(\frac{n}{A} \log A)$ height and $\mathcal{O}(\frac{A \log n}{\log A})$ width with $A \in \{2, \ldots, n\}$. *Subtree separation* means that in the drawing there are non-overlapping rectangles for every pair of disjoint subtrees. Observe that each choice of $2 \leq A \leq n$ implies an $\mathcal{O}(n \log n)$ drawing area. Now we consider the 8-grid. Due to the linear dependency among the different edge directions on the 8-grid, it is not possible to reach the result with straight-line edges.

Theorem 4 *Let T be an arbitrary ternary tree with n vertices. There is a planar upward drawing on the 8-grid of T with $\mathcal{O}(\frac{n}{A} \log A) \times \mathcal{O}(\frac{A \log n}{\log A})$ area, where $A \in \{2, \ldots, n\}$, and in total with less than $n - |\operatorname{leaves}(T)|$ bends.*

Proof: Let $T_3 = (V_3, E_3)$ be an arbitrary ternary tree with n vertices. First, we transform T_3 to a binary tree $T_2 = (V_2, E_2)$ as follows. For each vertex $v \in V_3$ with three children v_1, v_2, v_3 we replace the edges (v, v_1) and (v, v_2) by a new

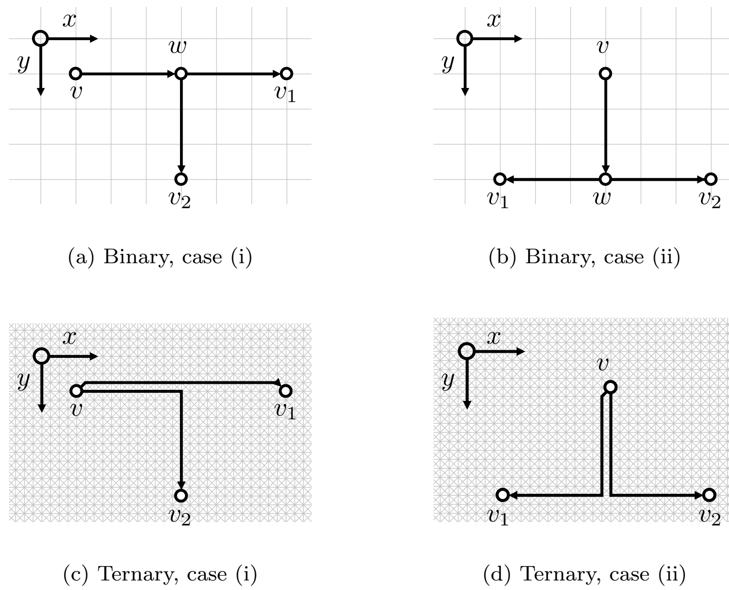


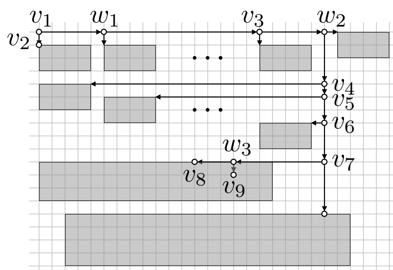
Figure 6: Ternarization

dummy vertex w and the edges (v, w), (w, v_1), and (w, v_2). Note that $|V_2|$ and $|E_2|$ are linear in n.

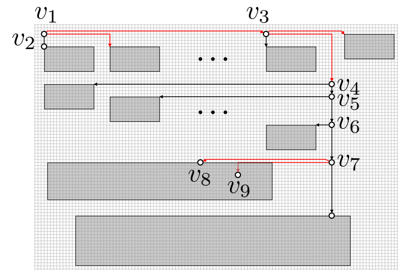
In the next step, we draw T_2 using the algorithm of Chan et al. [8, 9] out of the box. We obtain a drawing $\Gamma(T_2)$ on the 4-grid with a height in $\mathcal{O}(\frac{n}{A} \log A)$ and a width in $\mathcal{O}(\frac{A \log n}{\log A})$, where $A \in \{2, \ldots, n\}$. Each vertex $v \in V_2$ with two children has the following property. Either, in case (i), one outgoing edge of v has the same direction as the incoming edge e of v and the other is orthogonal to e, or, in case (ii), both outgoing edges of v are orthogonal to e. See Figs. 6(a) and (b).

Now, we transform the straight-line drawing $\Gamma(T_2)$ on the 4-grid into a drawing $\Gamma(T_3)$ on the 8-grid with bends. All non-dummy vertices are placed on identical coordinates in $\Gamma(T_3)$. We replace each dummy vertex $w \in V_2$ and its incident edges (v, w), (w, v_1), and (w, v_2) by the original edges (v, v_1) and (v, v_2) and determine their bend coordinates. All other (straight-line) edges remain identical in $\Gamma(T_3)$.

First, consider case (i). Let (x_u, y_u) be the coordinates of a vertex u in $\Gamma(T_2)$. W.l.o.g. an edge $(v, w) \in E_2$ for each dummy vertex w is directed to the east in $\Gamma(T_2)$, (w, v_1) has the same direction, and (w, v_2) points to the south, see Fig. 6(a). Then, $y_v = y_w = y_{v_1}$ and $x_w = x_{v_2}$. In $\Gamma(T_3)$ we add (v, v_1) with two bends at $(x_v + 0.25, y_v - 0.25)$ and $(x_{v_1} - 0.25, y_{v_1} - 0.25)$. We add (v, v_2) with one bend at (x_w, y_w) instead of the dummy vertex w. For an



(a) Recursive winding on the 4-grid



(b) Recursive winding on the 8-grid with less than n bends

Figure 7: Transformation of the binary tree drawing with dummy vertices w_1, w_2, and w_3 to the corresponding drawing of the ternary input tree

example see Fig. 6(c). It remains case (ii), where w. l. o. g. the edge (v, w) points to the south, (w, v_1) to the west, and (w, v_2) to the east. Here, we place the bends of the edge (v, v_1) at coordinates $(x_v - 0.25, y_v + 0.25)$ and $(x_w - 0.25, y_w)$. For (v, v_2) we introduce a bend at (x_w, y_w) instead of w. For an example see Figs. 6(b) and (d).

For all remaining directions of the incoming edge of w the construction is analogous. Note that the construction steps preserve planarity and subtree separation. $\Gamma(T_3)$ is a drawing with vertices at integral coordinates and some bends at multiples of 0.25. Quadrupling all coordinates transforms bends onto the grid while the area grows by a factor of 16. Thus, the asymptotic area bounds of Chan et al. [8, 9] hold. The number of introduced bends in total is less than $2\frac{n}{3} + \frac{n}{3} = n$ since one third of the edges gets two bends and another third one. Further, as no edge incident to a leaf needs a bend, we have less than $n - |\operatorname{leaves}(T_3)|$ bends. For a schematic example see Fig. 7. □

The construction of the proof can also be used for planar non-upward drawings of ternary trees on the 8-grid [8, 9]. Then, all directions of the 8-grid are used for the edge routing.

Corollary 2 *Let T be an arbitrary ternary tree with n vertices. There is a planar drawing on the 8-grid of T with $\mathcal{O}(\frac{n}{A}\log A) \times \mathcal{O}(A\log n + A)$ area, where $A \in \{2, \ldots, n\}$, and in total with less than $n - |\operatorname{leaves}(T)|$ bends.*

Choosing $A = \lceil \log n \rceil$ delivers a drawing with height $\mathcal{O}(\frac{n}{\log n}\log\log n)$, width $O(\log n)$, and, hence, area in $\mathcal{O}(n \log\log n)$.

5 $\mathcal{NP}$-hardness Results

In this section we present $\mathcal{NP}$-hardness results for planar unordered tree drawings on k-grids. There is always an ULP_k-drawing $\Gamma(T)$ of a d-ary tree T on

Table 2: Drawing with unit edge length is $\mathcal{NP}$-hard

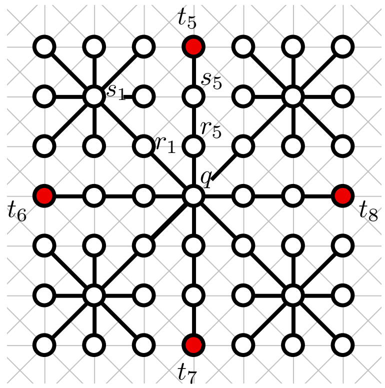
	$k = 4$	$k = 6$	$k = 8$
U_k	[3]	Theorem 6	Theorem 5
UL_k	Theorem 6	Theorem 6	Theorem 6
UP_k	Corollary 3	Conjecture 1	Corollary 3
ULP_k	Corollary 3	Conjecture 1	Corollary 3

the k-grid with $d < k$. A possible construction is similar to the drawing style of the complete 7-ary tree in Sect. 3. We set the lengths of the outgoing edges of a vertex u to $3^{height(T)-depth(u)-1}$ and then proceed top-down. For each vertex u we draw its $j < k$ outgoing edges in an arbitrary order with these lengths and with arbitrary directions. Similar to [1,3], we first shall restrict ourselves to the problem of drawing with unit edge length. Table 2 summarizes that drawing trees with unit edge length on a grid is in all listed cases $\mathcal{NP}$-hard. Afterwards, we consider the area complexity of these drawings without the unit edge length constraint.

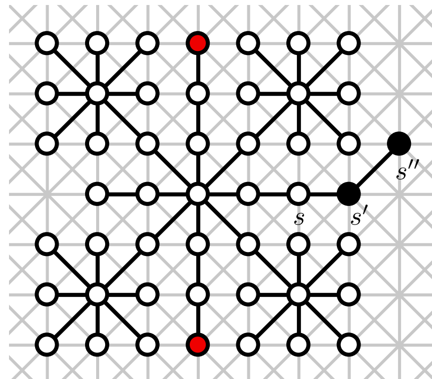
5.1 Unit Edge Length

We consider the complexity of constructing U_k-, UP_k-, UL_k- and ULP_k-drawings with unit edge lengths. First we show an $\mathcal{NP}$-hardness result for U_k-drawings, where we extend the results of the 4-grid [3, 12, 17] to trees of degree 8 on the 8-grid. This result should be adaptable, such that it holds also for binary trees on the 8-grid similar to [17]. 3-satisfiability (3-SAT) is the classical $\mathcal{NP}$-complete decision problem whether or not a given a Boolean expression E in 3-conjunctive normal (3-CNF) with n variables and c clauses has a satisfying assignment such that E evaluates to true [14]. 3-CNF demands that the *clauses* are logically and-conjuncted and that each clause contains exactly three literals which are conjuncted by logical ors. A *literal* is a Boolean variable which may be negated or not. We reduce the $\mathcal{NP}$-hard NOT-ALL-EQUAL-3-SAT (NAE-3-SAT) [14] by constructing a tree of degree 8 in polynomial time for the given expression E. NAE-3-SAT is a specialized variant of 3-SAT where only Boolean variable assignments are accepted which additionally cause at least one true and at least one false literal in each clause.

For a simple description, we use a free undirected tree in the following constructions. We define a *full tree* consisting of a vertex q with eight neighbors $r_1, \ldots, r_8$, see Fig. 8(a). In turn, these have incident edges $(r_1, s_1), \ldots, (r_8, s_8)$. The four vertices $s_1, \ldots, s_4$ in $\{s_1, \ldots, s_8\}$ additionally have seven adjacent vertices, called *corner leaves*. Each of the remaining four vertices $s_5, \ldots, s_8$ has only one additional neighbor, called *center leaf*. In Fig. 8(a) the leaves $t_5, \ldots, t_8$ are center leaves and all remaining leaves are corner leaves. We identify the *position* of a full tree by the coordinates of vertex q.



(a) Full tree



(b) Encode tree and striker (s, s', s'')

Figure 8: Full and encode trees on the 8-grid

Lemma 1 *Full trees have exactly one drawing with unit edge length on the 8-grid up to translation and labeling of the vertices.*

Proof: As mentioned above let $s_1, \ldots, s_4$ be the neighbors of the corner leaves and let the remaining $s_5, \ldots, s_8$ be the neighbors of the center leaves. Assume for contradiction that an edge (q, r_i) with $i \in \{1, \ldots, 4\}$ is drawn horizontally with length 1. As required, the seven other incident edges of q are also drawn with length 1. Then, there remain three possible edge directions for (r_i, s_i). One is horizontal and two are diagonal. If (r_i, s_i) is drawn horizontally, then the seven adjacent leaves of s_i cannot be placed satisfying unit edge length without an overlap. The same is true if (r_i, s_i) is drawn diagonally with unit edge length. Thus, (q, r_i) cannot be drawn horizontal. The same arguments hold for a vertical (q, r_i). Thus, (q, r_i) must be diagonal.

There remain five possible directions for the edge (r_i, s_i). Assume for contradiction that (r_i, s_i) has a different slope as (q, r_i). Then, the seven neighbors of s_i overlap with the neighbors of q. Therefore, the square containing s_i and its neighbors must be drawn at a corner of the overall 7×7 square S of grid points. No vertex can be placed outside of S, as otherwise there is an edge which is longer than 1. Finally, for the paths from q to the center leaves only the horizontal and vertical directions remain. □

Let an *encode tree* be a full tree omitting two center leaves, see Fig. 8(b) (without s' and s''). We call the two new leaves, i. e., the former neighbors of the omitted center leaves, *striker leaves*. Later we shall extend certain encode trees by *strikers* which are paths of length two added at a striker leaf, e. g., (s, s', s'') in Fig. 8(b). Consider a drawing of an encode tree with a given position of s. Then, the position of s' is predetermined. For s'' three possible grid points remain satisfying unit edge length. These lie outside the 7×7 square.

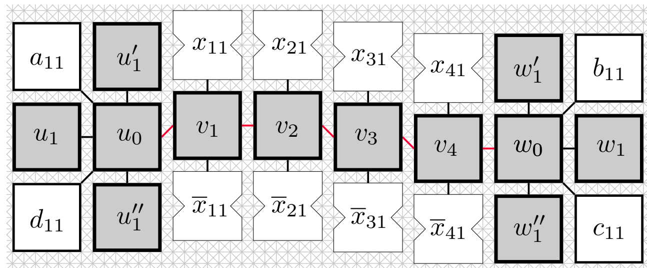


Figure 9: Basic tree $S(n, 1)$ with $n = 4$

We connect two full and encode trees inserting either an edge between center leaves, called *center connection*, or an edge between corner leaves, called *corner connection*. Note that in the remainder these two connection types are the only connections used between these two tree types. When it is obvious, we coarsen our view and use the notions vertex, leaf and path identifying full and encode trees as (meta-)vertices. Let u and v be center connected full trees. By Lemma 1, there are four *relative positions* for them in a drawing, i. e., u is *left of*, *right of*, *above* or *below* v. Let v have an absolute grid position and u a relative position to v. For u there are three possible grid positions satisfying edge length 1, e. g., if u is left of v, their y-coordinates differ by at most 1.

For a Boolean expression E in 3-CNF with c clauses and n variables we construct a tree $S(n, c + 1)$ of degree 8, see Fig. 10 ignoring the strikers. Initially, we introduce the *basic tree* $S(n, 1)$ containing the *basic spine* of center connected full trees $(u_0, v_1, \ldots, v_n, w_0)$, see Fig. 9. We add two center connected encode trees x_{i1} and $\overline{x}_{i1}$ as additional neighbors to each of the full trees v_i with $i \in \{1, \ldots, n\}$. Finally, we append three additional center connected full trees u_1, u'_1, u''_1 (w_1, w'_1, w''_1) and two corner connected full trees a_{11} and d_{11} (b_{11} and c_{11}) to the full tree u_0 (w_0). We denote a corner connected full tree α_{11} with $\alpha \in \{a, b, c, d\}$.

In the inductive step $j \to j + 1$ we expand $S(n, j)$ to $S(n, j + 1)$ by appending the full trees $u_{j+1}, u'_{j+1}, u''_{j+1}, w_{j+1}, w'_{j+1}, w''_{j+1}$ to $u_j, u'_j, u''_j, w_j, w'_j, w''_j$ via center connections. Again using center connections, we add the encode trees $x_{i,j+1}$ and $\overline{x}_{i,j+1}$ to x_{ij} and $\overline{x}_{ij}$, respectively. For each $k \in \{1, \ldots, j\}$ and $\alpha \in \{a, b, c, d\}$ we add the full tree $\alpha_{j+1,k}$ ($\alpha_{k,j+1}$) to the leaf α_{jk} (α_{kj}) using a center connection. Finally, we append the new full trees $\alpha_{j+1,j+1}$ to α_{jj} by corner connections.

We apply one additional construction step to the so far obtained tree $S(n, c)$ to frame it. This is done similarly to the inductive step from the previous paragraph, but using full trees instead of encode trees. We obtain the tree $S(n, c+1)$. This ensures that the free positions of each encode tree are restricted to the same y-coordinate and, then, as we will see later, strikers can only be embedded on the left or the right side of the encode trees for the c-th clause and

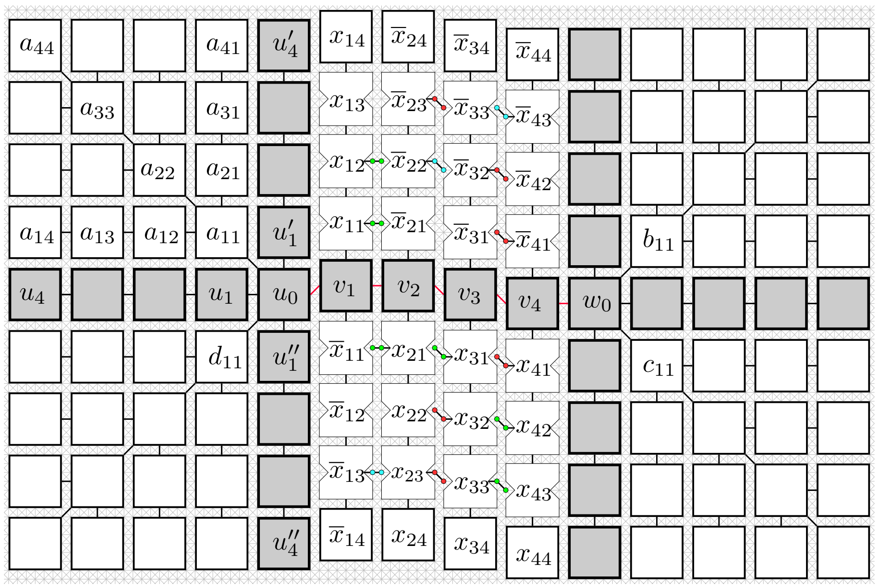


Figure 10: $T(E)$ of $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_1 \vee x_3 \vee \overline{x}_4)$ ($n = 4$ and $c = 3$) and assignment $x_1 = T, x_2 = x_3 = x_4 = F$

not to the north or to the south. In $S(n, c+1)$ we call the path $(x_{i,c+1}, \ldots, \overline{x}_{i,c+1})$ with $i \in \{1, \ldots, n\}$ the *i-th column* C_i.

Lemma 2 *Let $S(n, c+1)$ be drawn with unit edge length on the 8-grid and let its basic spine be $(v_0 = u_0, v_1, \ldots, v_n, w_0 = v_{n+1})$. Then, all vertices v_i with $i \in \{0, \ldots, n\}$ share the same relative position to their successor v_{i+1}, which is either left of, right of, above, or below.*

Proof: Considering the basic spine, the vertex u_0 has four center connected neighbors u_1, u_1', u_1'', v_1. W. l. o. g. v_1 is placed to the right of u_0 and u_1, u_1', u_1'' may be positioned left of, above, and below u_0, see Fig. 9. Assume for contradiction that the center connected neighbor v_2 is placed below (above) v_1. Each of the two full trees v_1 and v_2 has four center connected neighbors. This leads to a contradiction because there is no space left to place the fourth neighbor of v_2 considering edge length 1. Hence, v_2 must be placed to the right of v_1. The same can be shown by an inductive argument for the remaining full trees of the basic spine. As a consequence, all these full trees are placed side by side and the y-coordinates differ between neighbors at most by 1. The center connected neighbors w_1, w_1', w_1'' of w_0 are placed symmetrically to the respective neighbors of u_0. See Fig. 10 for an example. □

The basic spine is *horizontally embedded* if all neighbors are positioned in a planar way relatively left and right of each other, else it is *vertically embedded*. Let $\Gamma(S(n, c+1))$ be a drawing of $S(n, c+1)$ with a horizontally embedded basic spine. Then, the basic spine separates $\Gamma(S(n, c+1))$ into two halfs, the *top half* and the *bottom half*. Due to the freedom to permute incident edges, either the path $(x_{i1}, \ldots, x_{i,c+1})$ is drawn in the top half and $(\overline{x}_{i1}, \ldots, \overline{x}_{i,c+1})$ in the bottom half of each column C_i, or vice versa. We call the paths $(u_0, \ldots, u_{c+1}), (u_0, \ldots, u'_{c+1}), (u_0, \ldots, u''_{c+1})$ the *u-spines* of $S(n, c+1)$. Analogously, we define *w-spines*.

Lemma 3 *Let $S(n, c+1)$ be drawn with unit edge length and let the basic spine be embedded horizontally on the 8-grid. Then, all other edges have determined slopes (directions).*

Proof: Let $\Gamma(S(n, 1))$ be a drawing of the basic tree $S(n, 1)$ where w. l. o. g. the basic spine $(u_0, v_1, \ldots, v_n, w_0)$ is horizontally embedded and u_0 is placed left of v_1. Let u' be placed left of u_0, u' above u_0, and u'' below u_0 (symmetrically for w). Each v_i with $i \in \{1, \ldots, n\}$ has two center connected encode trees x_{i1} and $\overline{x}_{i1}$, which must be drawn above and below, respectively (or vice versa). So far all center connected full and encode trees have relative positions. Due to unit edge length, the horizontal grid distance between the corner connected full trees α_{11} with $\alpha \in \{a, b, c, d\}$ above (below) the basic spine is at most $7(n+2)+1$, which corresponds to $7(n+2)$ free grid points. As each full tree horizontally covers 7 grid points, the horizontal row of n encode trees cover in sum $7(n+2)$ points. The positions of these encode trees and of the corner connected full trees are fix. Analogously, the positions of the full trees vertically between the corner connected trees α_{11} are fix. Hence, all edges of the basic tree $S(n, 1)$ despite the center connections of the basic spine have a fix slope. The same argumentation can be applied in the induction step $j \to j+1$. $\square$

W. l. o. g. let the basic spine always be horizontally embedded to the right in the remainder. Then, in each column C_i with $i \in \{1, \ldots, n\}$ the center connection edges of the path $(x_{i,c+1}, \ldots, \overline{x}_{i,c+1})$ are drawn vertically. Each surrounding 7×7 square of an encode tree covers exactly two grid points which are not occupied by the encode tree and which have identical y-coordinates. We use these free grid points to encode the remaining information of the Boolean expression E into the tree $S(n, c+1)$.

Consider the i-th variable x_i and the j-th clause with $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, c\}$. If x_i does not appear in clause j, we append the striker $(s_{ij}, s'_{ij}, s''_{ij})$ to the encode tree x_{ij} and the striker $(\overline{s}_{ij}, \overline{s}'_{ij}, \overline{s}''_{ij})$ to $\overline{x}_{ij}$. If x_i occurs not negated in clause j, we add the striker $(s_{ij}, s'_{ij}, s''_{ij})$ to the encode tree x_{ij}. Finally, if the variable x_i occurs negated in clause j, we add the striker $(\overline{s}_{ij}, \overline{s}'_{ij}, \overline{s}''_{ij})$ to $\overline{x}_{ij}$. In the following, $T(E)$ identifies this extension of $S(n, c+1)$. Note that in a unit edge length drawing $\Gamma(T(E))$ all edges despite edges connecting the basic spine and the rear striker edges (s'_{ij}, s''_{ij}) or $(\overline{s}'_{ij}, \overline{s}''_{ij})$ have determined slopes.

Consider the encode tree z_{ij} with $z_{ij} \in \{x_{ij}, \overline{x}_{ij}\}$ and its striker $S = (s, s', s'')$ in $\Gamma(T(E))$. Since there is the freedom of vertically mirroring z_{ij}, S can be drawn either on its left or on its right side. According to Lemma 2, the y-coordinates of the columns C_i and C_{i-1} or C_{i+1} with $i \in \{2, \ldots, n-1\}$ differ at most by 1. However, if there is a free grid point at the right side of $z_{i-1,j}$ or the left side of

$z_{i+1,j}$, respectively, the vertex s'' can be embedded on it. For an example see the encode tree $\overline{x}_{22}$ in Fig. 10 with its striker S embedded to the right side. Note that a striker starting from an encode tree of C_1 (C_n) can only be embedded to the right side (left side).

Lemma 4 *Let E be a Boolean expression in 3-CNF with c clauses and n variables. E is satisfiable with at least one true and one false literal per clause if and only if there is a drawing $\Gamma(T(E))$ with unit edge length on the 8-grid.*

Proof: "⇒": Let $\tau(E)$ be a satisfying assignment for the Boolean expression E with n variables and c clauses. Compute the tree $T(E)$ as described above. To obtain a planar drawing $\Gamma(T(E))$ with unit edge length we have to determine, whether a path $(x_{i1}, \ldots, x_{i,c+1})$ will be embedded in the top half and its companion path $(\overline{x}_{i1}, \ldots, \overline{x}_{i,c+1})$ in the bottom half, or vice versa. If the variable x_i with $i \in \{1, \ldots, n\}$ is true, then embed the corresponding path $(x_{i1}, \ldots, x_{i,c+1})$ of column C_i in the top half, and in the bottom half otherwise. This is always possible as $\tau(E)$ ensures that each clause has at least one true and at least one false literal. This fits exactly to the fact that in the j-th row in the top (bottom) half of the drawing for the j-th clause at most two strikers can be embedded in a planar way as each encode tree can be vertically flipped. All other $n-3$ holes are occupied by strikers for variables not existing in clause j.

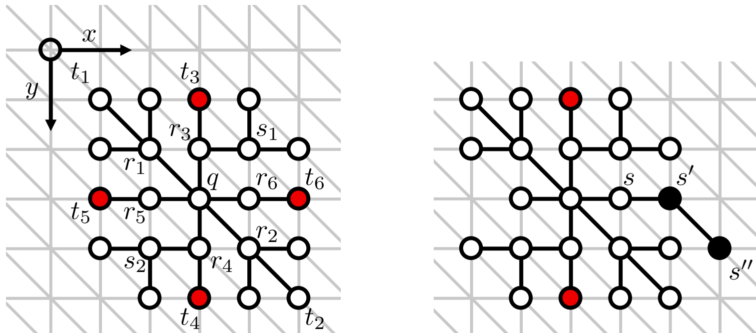
"⇐": Let $\Gamma(T(E))$ be a drawing with unit edge length of $T(E)$. According to Lemma 3 all edges have a determined slope despite the edges connecting the basic spine and the rear striker edges. Without strikers there are $n-1$ *holes*, i. e., two adjacent free grid points, in the j-th row with $j \in \{1, \ldots, c\}$ between neighbored encode trees in the top half and $n-1$ holes in the bottom half. In each row j (top and bottom half) we added $n-3$ strikers for the non existing variables in clause j. Therefore, in the top (bottom) half at most two more strikers can be placed in the j-th row. For negated and not negated literals in a clause we added in total three strikers. It is not possible to place all three strikers in the top (bottom) half. In a planar drawing with unit edge length there must be two strikers in the top half and the other in the bottom half, or vice versa.

A literal y_k with $k \in \{1, \ldots, n\}$ is either the variable x_k or its negation $\overline{x}_k$. Let y_k be in clause j. If y_k is not negated, then the literal is true if the corresponding striker $(s_{kj}, s'_{kj}, s''_{kj})$ is embedded in the top half of the drawing $\Gamma(T(E))$. Otherwise, if y_k is negated, then the literal is false if the striker $(s_{kj}, s'_{kj}, s''_{kj})$ is embedded in the top half. Hence, we obtain a satisfying assignment $\tau'(E)$ with respect to NAE-3-SAT with at least one literal true and at least one literal false in each clause. □

We obtain directly from Lemma 4 the following.

Theorem 5 *Let T be a 7-ary tree. Deciding whether or not there exists an U_8-drawing $\Gamma(T)$ with unit edge length is $\mathcal{NP}$-hard.*

The proof idea of the analogous result on the 6-grid is essentially the same as the above for the 8-grid. However, as there are technical differences and traps in



(a) Full tree on the 6-grid

(b) Encode tree and striker (s, s', s'')

Figure 11: Full and encode trees on the 6-grid

the constructions due to the one missing diagonal, w. l. o. g. from south-east to north-west on the grid, we nevertheless present the proof (where necessary) in the same detail to ensure correctness. The differences manifest especially in the north-west and south-east regions of the (drawings of) full trees, encode trees, and $T(E)$. For example and in contrast to Fig. 10, for the latter there is no option in this regions to corner connect the full trees b_{jj} and d_{jj}, $1 \leq j \leq c+1$, to enforce the rigidness of a drawing of $T(E)$. Note that the $\mathcal{NP}$-hardness was already claimed in [1], however, the detailed proof is presented for the first time in the following.

First, we extend the definition of a *full tree* for the 6-grid, see Fig. 11(a). The vertex q has six adjacent vertices $r_1, \ldots, r_6$. These have additional neighbors: r_1 (r_2) is adjacent to three leaves of which one is t_1 (t_2). r_3 (r_4) is adjacent to leaf t_3 (t_4) and to the inner vertex s_1 (s_2). r_5 (r_6) is adjacent to one leaf t_5 (t_6). There are no further vertices or edges. Analogously to the 8-grid, we call the adjacent leaves of r_1 and r_2 corner leaves and $t_3, \ldots, t_6$ center leaves.

Lemma 5 *Full trees have exactly one drawing on the 6-grid with unit edge length and center leaves on the boundary of a bounding* 5×5 *square (in grid points) up to swapping paths from* q *to the center leaves, translation, and labeling of the vertices.*

Proof: Let the root vertex q of a full tree be placed in the center of a 5×5 bounding square S (in grid points), w. l. o. g. at $(0, 0)$. Then, $r_1, \ldots, r_6$ are placed around q such that the edges $(q, r_1), \ldots, (q, r_6)$ have length 1.

Assume for contradiction that (q, r_1) is drawn to the east with edge length 1. Then, the adjacent leaves of r_1 occupy at least one of the grid positions $(1, 1)$ and $(1, -1)$ and the remaining adjacent vertices of q cannot be placed. With

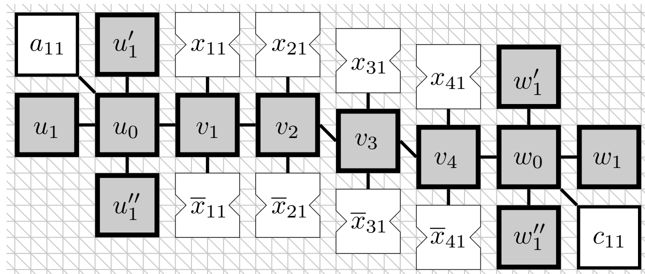


Figure 12: Basic tree $S(n, 1)$ with $n = 4$

symmetric arguments (q, r_1) cannot be drawn to the west or vertically. The same holds for the edge (q, r_2). Thus, (q, r_1) and (q, r_2) can only lie on the sole diagonal through q. As the precondition demands that each center leaf is on the boundary of S, each of them has to be placed in the middle of one boundary side. Then, there is not enough room to embedd both subtrees $T(r_3)$ and $T(r_4)$ induced by r_3 and r_4, respectively, together in the north-east or south-west quadrant. Thus, $T(r_3)$ is in the north-east and $T(r_4)$ in the south-west, or vice versa. Then, the positions of the remaining vertices are predetermined. □

An *encode tree* on the 6-grid is a full tree omitting the leaves t_5 and t_6, see Fig. 11(b) without s' and s''. Again, we call a path with length 2 starting at one of the two arising leaves a *striker*. In the example, the striker (s, s', s'') is appended to the *striker leaf* $r_6 = s$.

Consider a drawing of an encode tree with unit edge length and the two center leaves t_3 and t_4 placed at the boundary of the surrounding 5×5 square and a fixed position of s. Then, the position of s' is predetermined. For the leaf s'' there remain two possible grid points, which are located outside of the 5×5 square.

In the following we connect full trees and encoding trees like on the 8-grid. For these connections it is necessary to place the center leaves on the boundary of the respective 5×5 squares as required by Lemma 5. As on the 8-grid, center connections enforce relative positions of the participating full or encoding trees.

The basic spine is defined analogously and the construction of the basic tree $S(n, 1)$ is done similar to the 8-grid. The only difference is, that we omit the corner connected full trees b_{11} and d_{11}, see Fig. 12 in comparison to Fig. 9. The tree $S(n, 2)$ is the extension of $S(n, 1)$. The encoding trees, the u- and w-spines, and the full trees a_{11} and c_{11} are extended as on the 8-grid. The difference is, that we append the full trees b_{11} to w_1 and d_{11} to u_1, see Fig. 13. We call the subtrees rooted at α_{11} with $\alpha \in \{a, c\}$ *α-trees* and the subtrees rooted at β_{11} with $\beta \in \{b, d\}$ *β-trees*.

The inductive construction step $j \to j + 1$ is also similar to the 8-grid. We append a corresponding new encoding tree to each existing encoding tree in $S(n, j)$. The extension of the α-trees is identical, too. However, the extension

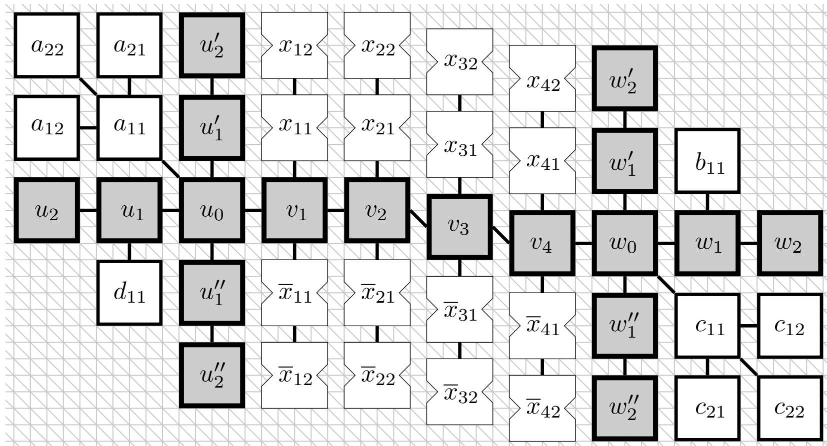


Figure 13: $S(n, 2)$ with $n = 4$

of the β-trees by new full trees varies due to the missing grid diagonal, which we describe in detail now. We center connect a new full tree $\beta_{j-\gamma+1,\gamma}$ to each existing full tree $\beta_{j-\gamma,\gamma}$ for $\gamma \in \{1, \ldots, j-1\}$ and a single new full tree $\beta_{1,j+1}$ to $\beta_{1,j}$. This construction step is repeated until $j + 1 = c$. Like on the 8-grid, we finally apply one additional construction step using full trees instead of encoding trees. Again, this ensures that the strikers later can only be embedded left or right of the encode trees of the c-th clause. Thus, for the tree construction on the 6-grid we obtain in analogy to Lemma 3 the following.

Lemma 6 *Let $S(n, c+1)$ be drawn with unit edge length and let the basic spine be embedded horizontally on the 6-grid. Then, all other edges have determined slopes (directions).*

Proof: The only difference to the 8-grid lies in the construction of the β-trees. Let $\Gamma(S(n, 1))$ be a drawing of the basic tree $S(n, 1)$ where w. l. o. g. the basic spine $(u_0, v_1, \ldots, v_n, w_0)$ is horizontally embedded and u_0 is placed left of v_1. Let u_1 be placed left of u_0, u'_1 above u_0, and u''_1 below u_0 (symmetrically for w). Each v_i with $i \in \{1, \ldots, n\}$ has two center connected encode trees x_{i1} and $\overline{x}_{i1}$, which must be drawn above and below, respectively, or vice versa. So far, all center connected full and encode trees have relative positions. Now, the corner connected full tree a_{11} (c_{11}) can only be drawn above (below) u_1 (w_1) and left of (right of) u'_1 (w'_1). Due to the unit edge length in $\Gamma(S(n, 1))$, the number of free grid points horizontally between u'_1 and w'_1 above the basic is at most $5n$. The same holds below the basic spine between u''_1 and w''_1. As each encode tree covers horizontally 5 grid points, the horizontal row of n encode trees covers in sum $5n$ grid points. Thus, the positions of these encode trees and of the corner connected full trees are fix. Hence, all edges of the basic tree $S(n, 1)$ up to the center connections of the basic spine have a determined slope, see Fig. 12.

In the first inductive step $1 \to 2$ the basic tree $S(n,1)$ is extended to $S(n,2)$, see Fig. 13. b_{11} is center connected to w_1 and d_{11} to u_1. The construction ensures that the position above w_1 is either occupied by b_{11} or w_2. The position below u_1 is analogously occupied by either d_{11} or u_2. Therefore, the full and encoding trees of each encode column C_i with $i \in \{1,\ldots,n\}$ are embedded vertically above each other, i. e., their x-coordinates are fixed. For the general inductive step $j \to j+1$ let w. l. o. g. w_0 be placed at the origin $(0,0)$ on a coarsened coordinate system where one unit covers 5 grid points. In case (i), b_{11} is then located at $(1,1)$. W. l. o. g. the w-spines $(w_0,\ldots,w_j)$, $(w_0,\ldots,w'_j)$, and $(w_0,\ldots,w''_j)$ are embedded horizontally to the east, vertically to the north, and vertically to the south, respectively. Now, we inductively prove that the "diagonal" positions $(j-\gamma,\gamma)$ and $(1,j)$ with $\gamma \in \{1,\ldots,j-1\}$ between w_j and w'_j are occupied by full trees of the b-tree. For $j=1$ the statement holds because in $S(n,1)$ there is no valid position between w_1 and w'_1. For $j=2$ in $S(n,2)$ there is one position occupied in the north-eastern direction by b_{11} or w_2. By induction the $j-1$ positions between w_j and w'_j are occupied by the full trees $b_{j-\gamma,\gamma}$ with $\gamma \in \{1,\ldots,j-1\}$. In the construction step we append the new full trees w_{j+1} and w'_{j+1} to the w-spines and center connect j new full trees $b_{j+1-\gamma,\gamma}$ with $\gamma \in \{1,\ldots,j-1\}$ to the existing full trees $b_{j-\gamma,\gamma}$ and $b_{1,j}$ to $b_{1,j-1}$. By induction there are $j-1$ occupied positions between w_j and w'_j. After applying the induction step there are exactly j positions between w_{j+1} and w'_{j+1}. In the remaining case (ii) b_{11} is located at the coarse coordinates $(2,0)$. Then, the path $(w_1,\ldots,w_j)$ is embedded vertically to the north and the b-tree is displaced one unit to the south-eastern direction. Then, the same argumentation as in case (i) shows that the diagonal positions are occupied. The proof for the d-tree is symmetric. Thus, together with the fix α-trees as shown in the proof of Lemma 3, the x-coordinates of the encoding trees in a drawing $\Gamma(S(n,c+1))$ are determined. Hence, all slopes between full and/or encoding trees are determined up to the center connections of the basic spine. □

With this in hands, we are now able to encode a Boolean expression E into the tree $S(n,c+1)$ to obtain $T(E)$ exactly as done on the 8-grid. Thus, the following holds by the same reasoning as in the proof of Lemma 4. For an example see Fig. 14.

Lemma 7 *Let E be a Boolean expression in 3-CNF with c clauses and n variables. E is satisfiable with at least one true and one false literal per clause if and only if there is a drawing $\Gamma(T(E))$ with unit edge length on the 6-grid.*

Together with the $\mathcal{NP}$-hardness results of Bhatt and Cosmadakis [3] on the 4-grid, the main result follows.

Theorem 6 *Let T be a d-ary tree with $d < k$ for $k \in \{4,6,8\}$. Deciding whether or not there exists an U_k-drawing $\Gamma(T)$ with unit edge length is $\mathcal{NP}$-hard.*

Now we restrict U_k-drawings using the aesthetic criteria local uniformity and patterns to obtain UL_k-, UP_k- and ULP_k-drawings. Remember, that these are

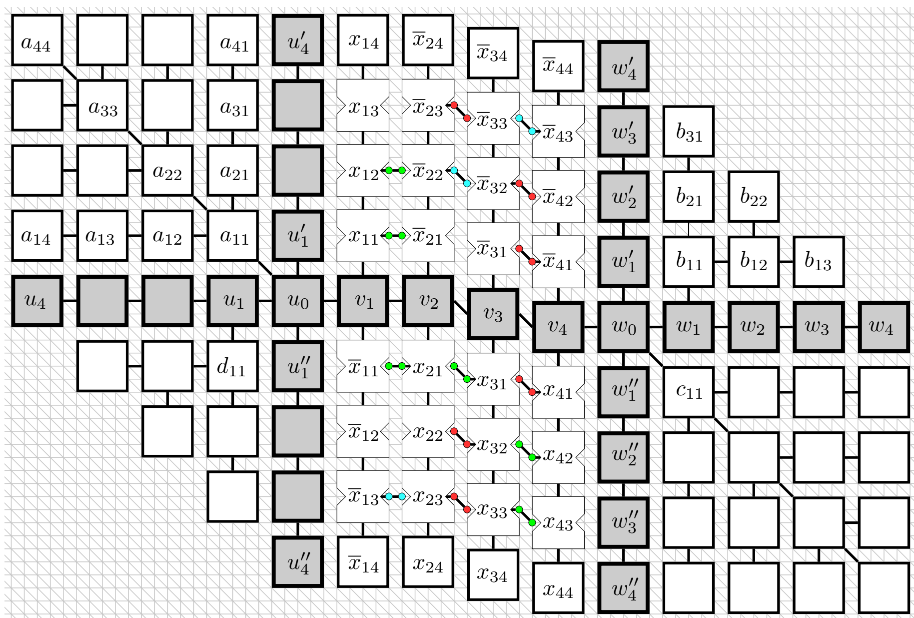


Figure 14: $T(E)$ of $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_1 \vee x_3 \vee \overline{x}_4)$ ($n = 4$ and $r = 3$) and assignment $x_1 = T, x_2 = x_3 = x_4 = F$

only defined for rooted trees. Unit edge length drawings trivially imply local uniformity and, thus, are automatically UL_k-drawings. Nevertheless, Theorem 6 also holds for UL_k-drawings with $k \in \{4, 6, 8\}$ and Corollary 3 for UP_k- and ULP_k-drawings with $k \in \{4, 8\}$.

Corollary 3 *Let T be a d-ary tree with $k \in \{4, 8\}$ and $d < k$. Deciding whether or not there exists an UP_k-drawing (ULP_k-drawing) $\Gamma(T)$ with unit edge length is $\mathcal{NP}$-hard.*

Proof: When using uniform slopes for the edges connecting the basic spine, the construction in the proof of Lemma 4 generates an ULP_k-drawing of $T(E)$ with unit edge length and $k \in \{4, 8\}$. □

For UP_6-drawings it is not possible to show the $\mathcal{NP}$-hardness with the presented approach because a full tree with its center leaves on the boundary is no valid pattern drawing. Nevertheless, we claim the following.

Conjecture 1 *Let T be a d-ary tree with $d < 6$. Deciding whether or not there exists an UP_6-drawing (ULP_6-drawing) $\Gamma(T)$ with unit edge length is $\mathcal{NP}$-hard.*

5.2 Area

Now we are interested in the area occupied by U_8-, UL_8-, UP_8- and ULP_8-drawings.

Proposition 1 *Let T be a tree with degree 8 and $A > 1$. Determining whether or not there exists an U_8-drawing $\Gamma(T)$ within area A is $\mathcal{NP}$-hard.*

Proof: (Sketch) Again we reduce from NAE-3-SAT. Let E be a Boolean expression with c clauses and n variables and let tree $T(E)$ of degree 8 be constructed as described in Sect. 5.1 with some additional edges. Let $j \in \{1, \ldots, c\}$. For each encode tree x_{1j} and $\overline{x}_{1j}$ in the first column C_1 we add new vertices b_{1j} and $\overline{b}_{1j}$ connected by the edges (s_{1j}, b_{1j}) and $(\overline{s}_{1j}, \overline{b}_{1j})$ to the striker leaf s_{1j} and $\overline{s}_{1j}$, respectively. To the last column C_n we add the edges (s_{nj}, b_{nj}) and $(\overline{s}_{nj}, \overline{b}_{nj})$ in the same way. Let $A = W \cdot H$ with $W = 7(2(c+1) + n + 2)$ and $H = 7(2(c+1) + 1)$. E is satisfiable with at least one true and one false literal per clause if and only if there is an U_8-drawing of $T(E)$ within area A.

"⇒": We argue similar to the if-part in the proof of Lemma 4. We draw the basic spine $(u_0, v_1, \ldots, v_n, w_0)$ horizontally on identical y-coordinates. We align the edges added to the striker leaves to the left in the first column C_1 and to the right in the last column C_n. Using the assignment $\tau(E)$ the strikers are aligned as described in Lemma 4. Then, $\Gamma(T(E))$ has total height H. Its total width corresponds to the sum of the lengths of a u-spine, the basic spine, and a w-spine which is W.

"⇐": Let $\Gamma(T(E))$ be a drawing of $T(E)$ within area A. The number of available grid points of area $A = W \cdot H$ with $W = 7(2(c+1) + n + 2)$ and $H = 7(2(c+1) + 1)$. The number of vertices in $T(E)$ is smaller by $2c$ than the number of available grid points in A. Therefore, for each clause j there are only two grid points left blank. This shall tighten the drawing in a row in the top or in the bottom half and therefore, the whole drawing, such that we can determine the assignment $\tau(E)$ analogously to the proof of Lemma 4. □

Corollary 4 *Let T be a 7-ary tree and $A > 1$. Determining whether or not there exists an UP_8, UL_8 or ULP_8-drawing $\Gamma(T)$ within area A is $\mathcal{NP}$-hard.*

Proof: First, consider ULP_8-drawings. Let $\Gamma(T(E))$ be a locally uniform pattern drawing within area $A = H \cdot W$, which is identical to the drawing in the proof of Proposition 1. Therefore, the remaining arguments can be applied analogously. The result also holds for UL_8- and UP_8-drawings because they are already ULP_8-drawings. □

The same statements for the k-grid with $k \in \{4, 6\}$ except for UP_6- and ULP_6-drawings shall be proven similarly. In the case $k = 6$, $\Gamma(T(E))$ must be extended to completely fill the remaining space in the north-east and the south-west of the surrounding rectangle with full trees.

6 Conclusion

We have shown the $\mathcal{NP}$-hardness for several problems of drawing trees with unit edge length on k-grids with $k \in \{4, 6, 8\}$. For the same types of drawings, we furthermore stated the $\mathcal{NP}$-hardness of using minimal area. For complete 7-ary trees on the 8-grid we presented a tight area bound of $\Theta(n^{\log_7 9})$ which is about $\Theta(n^{1.129})$ and for complete ternary trees we gave an almost linear upper bound of $\mathcal{O}(n^{1.048})$ for the needed area. Using bends allows drawing complete ternary trees in $\mathcal{O}(n)$ area. For arbitrary ternary trees we showed an upper area bound of $\mathcal{O}(n \log \log n)$. Each of the two latter algorithms generates in total less than n edge bends.

Future work is to complete the needed area bounds for arbitrary trees on the k-grid with or without allowing bends. We conjecture that ternary trees can be drawn upwards on the 8-grid with recursive winding using less than $\frac{n}{3}$ bends in $\mathcal{O}(n \log n)$ area and in $\mathcal{O}(n \log \log n)$ area without a common direction. For that, further optimizations in the subtree placement within the transformation of the intermediate binary to the final ternary tree drawing of Sect. 4.2 should be helpful. Formal $\mathcal{NP}$-hardness proofs for the unit edge length and minimal area UP_6- and ULP_6-drawings are missing up to now. It would be nice to have similar but more general results on bounds and $\mathcal{NP}$-hardness independent of a concrete k.

Acknowledgements

We thank Franz J. Brandenburg and Wolfgang Brunner for fruitful discussions and valuable hints.

References

[1] C. Bachmaier, F. J. Brandenburg, W. Brunner, A. Hofmeier, M. Matzeder, and T. Unfried. Tree drawings on the hexagonal grid. In I. G. Tollis and M. Patrignani, editors, *Proc. Graph Drawing, GD 2008*, volume 5417 of *LNCS*, pages 372–383. Springer, 2009. doi:10.1007/978-3-642-00219-9.

[2] C. Bachmaier and M. Matzeder. Drawing unordered trees on k-grids. In M. S. Rahman and S.-i. Nakano, editors, *Proc. Workshop on Algorithms and Computation, WALCOM 2012*, volume 7157 of *LNCS*, pages 198–210. Springer, 2012. doi:10.1007/978-3-642-28076-4.

[3] S. N. Bhatt and S. S. Cosmadakis. The complexity of minimizing wire lengths in VLSI layouts. *Inf. Process. Lett.*, 25(4):263–267, 1987. doi:10.1016/0020-0190(87)90173-6.

[4] A. Bloesch. Aestetic layout of generalized trees. *Softw. Pract. Exper.*, 23(8):817–827, 1993. doi:10.1002/spe.4380230802.

[5] F. J. Brandenburg. Nice drawings of graphs are computationally hard. In P. Gorny and M. Tauber, editors, *Vis. Human-Comp. Interact.*, volume 439 of *LNCS*, pages 1–15. Springer, 1990. doi:10.1007/3-540-52698-6.

[6] W. Brunner and M. Matzeder. Drawing ordered $(k-1)$-ary trees on k-grids. In U. Brandes and S. Cornelsen, editors, *Proc. Graph Drawing, GD 2010*, volume 6502 of *LNCS*, pages 105–116. Springer, 2011. doi:10.1007/978-3-642-18469-7.

[7] C. Buchheim, M. Jünger, and S. Leipert. Improving Walker's algorithm to run in linear time. In M. Goodrich and S. Kobourov, editors, *Proc. Graph Drawing, GD 2001*, volume 2528 of *LNCS*, pages 344–353. Springer, 2002. doi:10.1007/3-540-36151-0.

[8] T. Chan, M. T. Goodrich, S. R. Kosaraju, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. In *Proc. Graph Drawing, GD 1996*, volume 1190 of *LNCS*, pages 63–75. Springer, 1997. doi:10.1007/3-540-62495-3.

[9] T. M. Chan, M. T. Goodrich, S. R. Kosaraju, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. *Comput. Geom. Theory Appl.*, 23(2):153–162, 2002. doi:10.1016/S0925-7721(01)00066-9.

[10] 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. doi:10.1016/0925-7721(92)90021-J.

[11] P. Eades. Drawing free trees. *Bull. Inst. Comb. Appl.*, 5:10–36, 1992.

[12] P. Eades and S. Whitesides. The logic engine and the realization problem for nearest neighbor graphs. *Theor. Comput. Sci.*, 169(1):23–37, 1996. doi:10.1016/S0304-3975(97)84223-5.

[13] F. Frati. Straight-line orthogonal drawings of binary and ternary trees. In S.-H. Hong, T. Nishizeki, and W. Quan, editors, *Proc. Graph Drawing, GD 2007*, volume 4875 of *LNCS*, pages 76–87. Springer, 2008. doi:10.1007/978-3-540-77537-9.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness*. W. H. Freeman & Co., 1990.

[15] A. Garg, M. T. Goodrich, and R. Tamassia. Planar upward tree drawings with optimal area. *Int. J. Comput. Geometry Appl.*, 6(3):333–356, 1996. doi:10.1142/S0218195996000228.

[16] A. Garg and A. Rusu. Straight-line drawings of binary trees with linear area and arbitrary aspect ratio. *J. Graph Alg. Appl.*, 8(2):135–160, 2004. doi:10.7155/jgaa.00086.

[17] A. Gregori. Unit-length embedding of binary trees on a square grid. *Inf. Process. Lett.*, 31(4):167–173, 1989. doi:10.1016/0020-0190(89)90118-X.

[18] G. Kant. Hexagonal grid drawings. In E. W. Mayr, editor, *Proc. Workshop on Graph Theoretic Concepts in Computer Science, WG 1992*, volume 675 of *LNCS*, pages 263–276. Springer, 1992. doi:10.1007/3-540-56694-5.

[19] M. Kaufmann and D. Wagner. *Drawing Graphs*, volume 2025 of *LNCS*. Springer, 2001. doi:10.1007/3-540-44969-8.

[20] M. R. Kramer and J. L. van Leeuwen. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. In F. P. Preparata, editor, *VLSI-Theory*, volume 2 of *Advances in Computing Research*, pages 129–146. JAI Press, 1984.

[21] K. Marriott and P. J. Stuckey. $\mathcal{NP}$-completeness of minimal width unordered tree layout. *J. Graph Alg. Appl.*, 8(3):295–312, 2004. doi:10.7155/jgaa.00093.

[22] M. Matzeder. *Zeichnen von Bäumen auf Gittern*. Dissertation, Fakultät für Informatik und Mathematik, Universität Passau, 2012.

[23] E. M. Reingold and J. S. Tilford. Tidier drawing of trees. *IEEE Trans. Software Eng.*, 7(2):223–228, 1981. doi:10.1109/TSE.1981.234519.

[24] C.-S. Shin, S. K. Kim, and K.-Y. Chwa. Area-efficient algorithms for straight-line tree drawings. *Comput. Geom. Theory Appl.*, 15(4):175–202, 2000. doi:10.1016/S0925-7721(99)00053-X.

[25] K. J. Supowit and E. M. Reingold. The complexity of drawing trees nicely. *Acta Inf.*, 18:377–392, 1982. doi:10.1007/BF00289576.

[26] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Computers*, 30(2):135–140, 1981. doi:10.1109/TC.1981.6312176.

[27] J. Q. W. Walker. A node-positioning algorithm for general trees. *Softw. Pract. Exper.*, 20(7):685–705, 1990. doi:10.1002/spe.4380200705.