Drawing Unordered Trees on k-Grids

Christian Bachmaier and Marco Matzeder

University of Passau, Germany
{bachmaier,matzeder}@fim.uni-passau.de

Abstract. We present almost linear area bounds for drawing complete trees on the octagonal grid. For 7-ary trees we establish an upper and lower bound of $\Theta(n^{1.129})$ and for ternary trees the bounds of $\mathcal{O}(n^{1.048})$ and $\Theta(n)$, where the latter needs edge bends. 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 and hexagonal grid to the octagonal grid.

1 Introduction

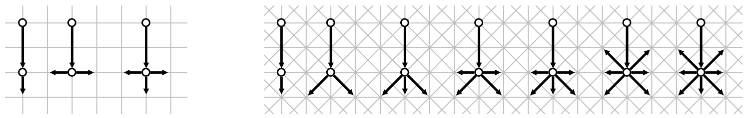
Trees are a fundamental data structure in computer science to represent hierarchies. Amongst others they are used as family trees in social networks or inheritance structures in UML-diagrams. Their visualization is an important field in graph drawing [7,9,13,14,21]. Often trees are unordered, e. g., 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 [17,20,22]: y-coordinates of the vertices correspond to their depth, centered parents over their children, minimal distance between vertices, integral coordinates, 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. [7]. Marriott and Stuckey [19] 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 [22] for order-preserving drawings. The common drawing algorithm for binary trees was introduced by Reingold and Tilford [20] and generalized to d-ary trees by Walker [3,6,24], 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 [2,7,8,11,13,14,21,23]. The 6- and the 8-grid with additional axes are used to draw trees with higher degree [1,5,16]. A motivation for such grids are discrete representations of radial drawings [9]. In our companion paper [5] 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 [2] 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 proved by the logic engine approach [10]. For binary trees this was shown by Gregori [15]. $\mathcal{NP}$-hardness results for minimum area were presented in [4, 18]. We presented an equivalent result for the 6-grid for trees with degree up to six [1]. In the plane only two grid axes are linear independent. This shall cause some problems which do not occur on the 4-grid for compacting drawings on higher order grids containing more than 2 axes. Furthermore, the degree of difficulty increases with the number of available directions.

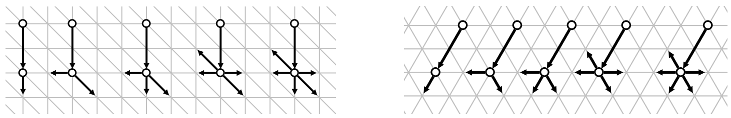
The remainder is organized as follows. After some definitions in Sect. 2 we present a tight area bound for drawing complete 7-ary trees in Sect. 3. Afterwards we show an almost linear upper area bound for straight-line drawings and a linear bound for drawings with bends of ternary trees on the 8-grid in Sect. 4. Finally, we show that it is $\mathcal{NP}$-hard to decide whether or not there is a unit edge length drawing for arbitrary trees with degree 8 and whether or not there is a drawing within a given area in Sect. 5.

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. We extend the 4-grid with its four directions to the *hexagonal* or *6-grid* [1, 5, 16] 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 $(k-1)$-ary tree $T = (V, E)$ on a k-grid 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 surrounding rectangle and the *aspect ratio* is the quotient of its height and its width. The following definitions of drawing styles are in accordance to [5], where we replace "O" for ordered by "U" for unordered. An U_k*-drawing* is a drawing of an unordered $(k-1)$-ary tree on a k-grid. A tree drawing is *locally uniform* if 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 $(k-1)$-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. For the various k-grids they are shown in



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



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

Fig. 1. Patterns on the k-grids

Fig. 1. 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 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 $\mathcal{O}(9^{\log_7 n})$ which is about $\mathcal{O}(n^{1.129})$.

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

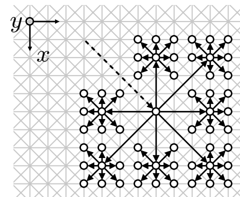


Fig. 2. Complete 7-ary tree

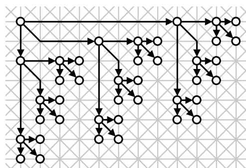


Fig. 3. Complete 3-ary tree with bends

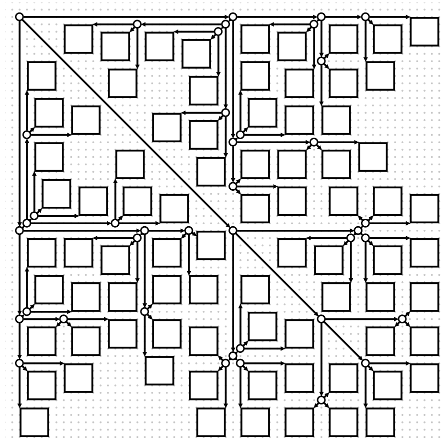


Fig. 4. Drawing scheme for Theorem 2 on the 8-grid (grid lines omitted)

least $3^{h-1} - 1$. Then one corner extreme point of T_5 overlaps the diagonal axis from r to north-east. Therefore, it is not possible to use this diagonal direction for T_6. The same can be shown for T_1, T_3, and T_7 by symmetrical arguments.

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 symmetrically). 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 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 symmetrical 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 Complete Ternary Trees

Each complete ternary tree can be drawn on the 4-grid within $\mathcal{O}(n^{1.262})$ area [11]. For strictly upward drawings on the 6-grid there is a tight bound of $\Theta(n^{1.262})$ [1]. 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 4 shows one recursion step for $i \to i + 4$ with $i \leq h$. Initially for $i = h \bmod 4 \in \{0, \ldots, 3\}$ we draw the tree $T(i)$ in a square with the root at the corner within constant area. 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 $S(i+4) = 10 \cdot S(i) + c \leq 10^{\lceil i/4 \rceil} + (c \cdot \sum_{i=0}^{\lceil i/4 \rceil} 10^i) < 10^{\lceil i/4 \rceil} + c \cdot 10^{\lceil i/4 \rceil + 1}$. 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 $\mathcal{O}(100^{(\log_3 n)/4}) = \mathcal{O}(n^{(\log_3 100)/4}) \subset \mathcal{O}(n^{1.048})$. □

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 (Sketch). The proof is done similar to the proof of Theorem 3 in [8] where either subtrees of even height are placed vertically and subtrees with odd height are placed horizontally, or vice versa, see Fig. 3. □

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 $(k-1)$-ary tree T on the k-grid. A possible construction is similar to the construction of the complete $(k-1)$-ary tree, see 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. First, similar to [1,2] we shall restrict ourselves to the problem of drawing with unit edge length and 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 [2,10,15] and the 6-grid [1] 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 [15]. We reduce *NOT-ALL-EQUAL-3-SAT* (*NAE3SAT*) [12] by constructing a tree of degree eight for a given Boolean expression E in *3-CNF* with n variables and c clauses in polynomial time.

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 neighbours $r_1, \ldots, r_8$, see Fig. 5. 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 one additional neighbor, called *center leaf*. In Fig. 5 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.

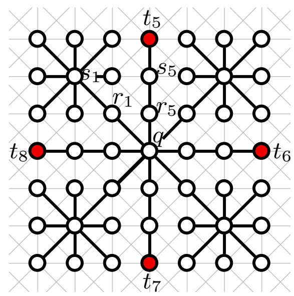


Fig. 5. Full tree

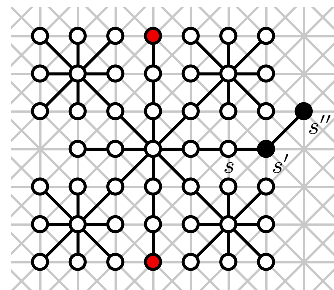


Fig. 6. Encode tree and striker (s, s', s'')

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 the 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 horizontal, 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, (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 whole 7×7 square of grid points. No vertex can be placed outside of the 7×7 square, as otherwise there is an edge longer than 1. Finally, for the paths from q to the center leaves only the horizontal and vertical directions remain. $\square$

Let an *encode tree* be a full tree omitting two center leaves, see Fig. 6 (without s' and s''). Later we shall extend certain encode trees by *strikers* which are paths of length two, added at one of the two new leaves (former parents of omitted center leaves), e. g., (s, s', s'') in Fig. 6. We call the two new leaves *striker leaves*. 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 outside the 7×7 square, which satisfy unit edge length.

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. Due to 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. For example, if u is left of v, their y-coordinates may differ by at most 1.

For a Boolean expression E in *3-CNF* with c clauses and n variables we construct a tree of degree 8 $S(n, c+1)$, see Fig. 7. Initially, we introduce the *basic tree* $S(n, 1)$ containing the *basic spine* of center connected full trees $(u_0, v_1, \ldots, v_n, w_0)$. Add two center connected encode trees x_{i1} and $\overline{x}_{i1}$ as additional neighbors to each of the full trees v_i with $i \in \{1, \ldots, n\}$. 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 with 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, the new full trees $\alpha_{j+1,j+1}$ are corner connected to α_{jj}.

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 $S(n, c+1)$. As we will see later, this ensures that the free positions of each encode tree are restricted to the same y-coordinate. 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. 7. Assume for contradiction that the center connected neighbour 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 neighbours at most by 1. The center connected neighbours w_1, w'_1, w''_1 of w_0 are placed symmetrically to the respective neighbors of u_0, see Fig. 7 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 em-

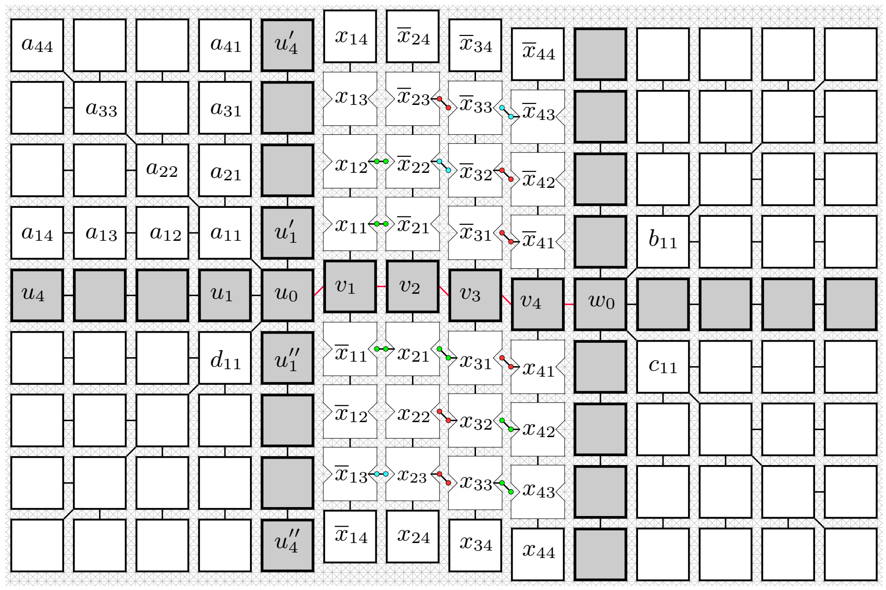


Fig. 7. $T(E)$ of $E = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_1 \vee x_3 \vee \overline{x}_4)$ ($n = 4$ and $c = 3$) and assignment $x_1 = T, x_2 = x_3 = x_4 = F$

bedded 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 the unit 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.

In $\Gamma(T(E))$ consider the encode tree z_{ij} with $z_{ij} \in \{x_{ij}, \overline{x}_{ij}\}$ and its striker $S = (s, s', s'')$. As 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 $z_{i-1,j}$ resp. the left side of $z_{i+1,j}$, the vertex s'' can be embedded on it. For an example see the encode tree $\overline{x}_{22}$ in Fig. 7 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. "$\Rightarrow$": 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 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.

"$\Leftarrow$": 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 $j \in \{1, \dots, c\}$ 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 $k \in \{1, \dots, n\}$ 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 *NAE3SAT* with at least one literal true and at least one literal false in each clause. □

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

Proof. For the 8-grid the correctness follows from Lemma 4. Bhatt and Cosmadakis [2] showed the $\mathcal{NP}$-hardness on the 4-grid. For the 6-grid see [1].

Now we restrict U_k-drawings using the aesthetics local uniformity and patterns and obtain UL_k-, UP_k- and ULP_k-drawings. Note that these are only defined for rooted trees. However, treating unit edge length in UL_k- and ULP_k-drawings is tedious since local uniformity is trivially given then. Nevertheless, Theorem 4 also holds for UL_k-drawings and Corollary 2 for ULP_k-drawings.

Corollary 2. *Let T be a $(k-1)$-ary tree. Deciding whether or not there exists an UP_k-drawing $\Gamma(T)$ with with unit edge length and $k \in \{4, 6, 8\}$ is $\mathcal{NP}$-hard.*

Proof. If using uniform slopes for the edges connecting the basic path, the construction in the proof Lemma 4 generates an UP_k-drawing of $T(E)$. □

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 *NAE3SAT*. 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 $j \in \{1, \dots, c\}$. 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 $(\bar{s}_{nj}, \bar{b}_{nj})$ 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.

"$\Rightarrow$": 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.

"$\Leftarrow$": 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. $\square$

Corollary 3. *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. $\square$

The same statements for the k-grid with $k \in \{4, 6\}$ shall be proven similarly.

6 Conclusion

We have shown the $\mathcal{NP}$-hardness for several problems of drawing trees on k-grids with unit edge length or on minimal area for U_k- and UL_k-drawings with $k \in \{4, 6, 8\}$ and UP_k- and ULP_k-drawings with $k \in \{4, 8\}$. 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.

Future work is to investigate the needed area for arbitrary trees on the k-grid with or without allowing edge bends. We conjecture, that ternary trees can be drawn upwards on the 8-grid using far 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 in analogy to recursive winding [7].

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

References

1. Bachmaier, C., Brandenburg, F.J., Brunner, W., Hofmeier, A., Matzeder, M., Unfried, T.: Tree drawings on the hexagonal grid. In: Tollis, I.G., Patrignani, M. (eds.) GD 2008. LNCS, vol. 5417, pp. 372–383 (2009)
2. Bhatt, S.N., Cosmadakis, S.S.: The complexity of minimizing wire lengths in VLSI layouts. Inf. Process. Lett. 25(4), 263–267 (1987)
3. Bloesch, A.: Aestetic layout of generalized trees. Softw. Pract. Exper. 23(8), 817–827 (1993)
4. Brandenburg, F.J.: Nice drawings of graphs are computationally hard. In: Gorny, P., Tauber, M. (eds.) Vis. Human-Comp. Interact. LNCS, vol. 439, pp. 1–15 (1990)
5. Brunner, W., Matzeder, M.: Drawing ordered (k - 1)-ary trees on k-grids. In: Brandes, U., Cornelsen, S. (eds.) GD 2010. LNCS, vol. 6502, pp. 105–116 (2011)
6. Buchheim, C., Jünger, M., Leipert, S.: Improving Walker's algorithm to run in linear time. In: Goodrich, M., Kobourov, S. (eds.) GD 2001. LNCS, vol. 2528, pp. 344–353 (2002)
7. Chan, T.M., Goodrich, M.T., Kosaraju, S.R., Tamassia, R.: Optimizing area and aspect ratio in straight-line orthogonal tree drawings. Comput. Geom. Theory Appl. 23(2), 153–162 (2002)
8. Crescenzi, P., Di Battista, G., Piperno, A.: A note on optimal area algorithms for upward drawings of binary trees. Comput. Geom. Theory Appl. 2, 187–200 (1992)
9. Eades, P.: Drawing free trees. Bull. Inst. Comb. Appl. 5, 10–36 (1992)
10. Eades, P., Whitesides, S.: The logic engine and the realization problem for nearest neighbor graphs. Theor. Comput. Sci. 169(1), 23–37 (1996)
11. Frati, F.: Straight-line orthogonal drawings of binary and ternary trees. In: Hong, S.H., Nishizeki, T., Quan, W. (eds.) GD 2007. LNCS, vol. 4875, pp. 76–87 (2008)
12. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)
13. Garg, A., Goodrich, M.T., Tamassia, R.: Planar upward tree drawings with optimal area. Int. J. Comput. Geometry Appl. 6(3), 333–356 (1996)
14. Garg, A., Rusu, A.: Straight-line drawings of binary trees with linear area and arbitrary aspect ratio. J. Graph Algo. Appl. 8(2), 135–160 (2004)
15. Gregori, A.: Unit-length embedding of binary trees on a square grid. Inf. Process. Lett. 31(4), 167–173 (1989)
16. Kant, G.: Hexagonal grid drawings. In: Mayr, E.W. (ed.) WG 1992. LNCS, vol. 675, pp. 263–276 (1992)
17. Kaufmann, M., Wagner, D.: Drawing Graphs, LNCS, vol. 2025. Springer (2001)
18. Kramer, M.R., van Leeuwen, J.L.: The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. In: Preparata, F.P. (ed.) VLSI-Theory, Advances in Computing Research, vol. 2, pp. 129–146. JAI Press (1984)
19. Marriott, K., Stuckey, P.J.: NP-completeness of minimal width unordered tree layout. J. Graph Algo. Appl. 8(3), 295–312 (2004)
20. Reingold, E.M., Tilford, J.S.: Tidier drawing of trees. IEEE Trans. Software Eng. 7(2), 223–228 (1981)
21. Shin, C.S., Kim, S.K., Chwa, K.Y.: Area-efficient algorithms for straight-line tree drawings. Comput. Geom. Theory Appl. 15(4), 175–202 (2000)
22. Supowit, K.J., Reingold, E.M.: The complexity of drawing trees nicely. Acta Inf. 18, 377–392 (1982)
23. Valiant, L.G.: Universality considerations in VLSI circuits. IEEE Trans. Computers 30(2), 135–140 (1981)

24. Walker, J.Q.W.: A node-positioning algorithm for general trees. Softw. Pract. Exper. 20(7), 685–705 (1990)