IFIG
RESEARCH
REPORT

INSTITUT FÜR INFORMATIK

GRID GRAPHS WITH DIAGONAL EDGES AND THE COMPLEXITY OF XMAS MAZES

Markus Holzer   Sebastian Jakobi

IFIG RESEARCH REPORT 1201
JANUARY 2012

Institut für Informatik
JLU Gießen
Arndtstraße 2
35392 Giessen, Germany
Tel: +49-641-99-32141
Fax: +49-641-99-32149
mail@informatik.uni-giessen.de
www.informatik.uni-giessen.de

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

IFIG Research Report
IFIG Research Report 1201, January 2012

Grid Graphs with Diagonal Edges and the Complexity of Xmas Mazes

Markus Holzer[1] and Sebastian Jakobi[2]

Institut für Informatik, Universität Giessen
Arndtstraße 2, 35392 Giessen, Germany

Abstract. We investigate the computational complexity of some maze problems, namely the reachability problem for (undirected) grid graphs with diagonal edges, and the solvability of Xmas tree mazes. Simply speaking, in the latter game one has to move sticks of a certain length through a maze, ending in a particular game situation. It turns out that when the number of sticks is bounded by some constant, these problems are closely related to the grid graph problems with diagonals. If on the other hand an unbounded number of sticks is allowed, then the problem of solving such a maze becomes PSPACE-complete. Hardness is shown *via* a reduction from the nondeterministic constraint logic (NCL) of Demaine and Hearn to Xmas tree mazes.

Categories and Subject Descriptors: F.1.3 [**Computation by Abstract Devices**]: Complexity Measures and Classes—*Reducibility and completeness*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Path and circuit problems*;

Additional Key Words and Phrases: constraint logic, grid graph reachability, puzzles

[1]E-mail: holzer@informatik.uni-giessen.de
[2]E-mail: jakobi@informatik.uni-giessen.de

Copyright © 2012 by the authors

1 Introduction

Spirals and rudimentary maze- or labyrinth-like forms can be already found on prehistoric cave walls all over the world. Nowadays, the interpretation of these ancient drawings is that they were believed to be passages to the underworld, but their true meaning is lost in the darkness of the ages. One of the most famous mazes (labyrinths) is that of King Minos of Crete, which was designed according to the Greek mythology by the artificer Daedalus to hold the Minotaur. Ovid, in his Metamorphoses, has mentioned that Daedalus constructed the labyrinth so cunningly that he himself could barely escape it after he built it. The Minotaur was eventually killed by Theseus, who escaped the labyrinth with the help of Ariadne's string, which is used to navigate the labyrinth. In modern terms these can be seen as an algorithm for searching a maze. Although the use of mazes and labyrinths in several contexts dates back to the beginning of mankind, even nowadays they haven't lost their attractiveness.

We investigate the computational complexity of different maze-like problems. First we consider the reachability problem for (undirected) grid graphs with diagonal edges, depending on whether one or both diagonal edges in a unit square are allowed. It turns out that the former problem can be reduced back to (undirected) grid graphs, while the latter is L-complete (NL-complete for directed grid graphs, respectively). Second, we study Xmas tree mazes, that became popular recently—see, for instance, `www.puzzlebeast.com/christmastree`. Here the player moves sticks of length ℓ through the corridors of a maze—for a precise definition of the game we refer to the appropriate section. The challenge is to move a stick from a starting position to a designated target position, which is not that easy, as it looks first. Several sticks may prevent certain moves, since they may block each other's movements. In fact, if the number of sticks in such a maze is not bounded by some constant, it turns out that solving such a maze is extremely complicated from a complexity theoretical point of view, namely PSPACE-complete. While containment in PSPACE is easy, the hardness is shown with the help of the recently introduced uniform framework for modeling games, the nondeterministic constraint logic (NCL) of Demaine and Hearn [3]. If the number of sticks in an Xmas tree maze is bounded, then it turns out that these problems are closely related to the reachability problems for undirected grid graphs with diagonal edges.

The paper is organized as follows: In the next section we introduce the necessary notations. Then in Section 3 we study the different reachability problems for (undirected) grid graphs with diagonal edges. These problems are useful for proving our results on Xmas tree mazes with a bounded number of sticks in Section 4.1. Section 4.2 shows that when the number of sticks is not bounded, the solvability problem for such mazes becomes PSPACE-complete. The results on Xmas mazes are summarized in Table 1 in the ultimate section, where also open problems and relations to some other maze puzzle problems are discussed.

2 Definitions

We assume familiarity with the basic concepts of complexity theory [5] such as the inclusion chain $\mathsf{AC}^0 \subset \mathsf{NC}^1 \subseteq \mathsf{L} = \mathsf{SL} \subseteq \mathsf{NL} \subseteq \mathsf{AL} = \mathsf{P} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE}$. Here AC^0 and NC^1 refer to the sets of problems accepted by polynomial size uniform families of Boolean {AND, OR, NOT}-circuits having, respectively, unbounded fan-in and constant depth, and bounded fan-in and logarithmic depth. L is the set of problems accepted by deterministic logarithmic space bounded Turing machines. SL and NL can be taken to be the sets of problems logspace-reducible to the undirected graph reachability (UGR) and to the directed graph reachability (GR) problems respectively and AL is the set of problems accepted by alternating logspace bounded Turing machines. P (NP, respectively) is the set of problems accepted by deterministic (nondeterministic, respectively) polynomial time bounded Turing machines and PSPACE is the set of problems accepted by deterministic or nondeterministic polynomial space bounded Turing machines. All the relationships depicted in the inclusion chain have been known for a quarter of a century, except for $\mathsf{L} = \mathsf{SL}$, shown in [6].

Two other particularly relevant problems are undirected grid graph reachability (UGGR) and constraint logic (CL). The former problem is defined as follows: given an $n \times n$ grid of nodes such that an edge only connects immediate vertical or horizontal neighbors, is there a path from node s to node t, where s and t are designated nodes from the grid? UGGR is NC^1-hard under AC^0 reducibility, it belongs to L, yet it is not known to be L-hard [1]. The latter problem, i.e., constraint logic or more precisely nondeterministic constraint logic (NCL), is defined as follows: given a constraint graph G and an edge e of G, is there a sequence of legal moves on G that eventually reverses e? Here a constraint graph is a directed graph with edge weights from the set $\{1, 2\}$ and where each vertex has a non-negative minimum inflow. Here the inflow of a vertex is the sum of the weights of inward-directed edges. A legal configuration of a constraint graph has an inflow of at least the minimum inflow at each vertex (these are the constraints that have to be satisfied), and a legal move on a constraint graph is the reversal of a single edge that results in a legal configuration. NCL is PSPACE-complete, even for planar constraint graphs built by AND- and OR-vertices only [3]—see Figure 1 for AND- and OR-vertices. Thus in order

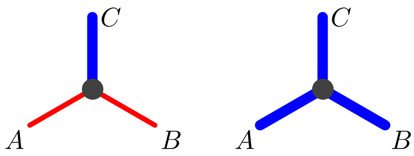

Fig. 1. Nondeterministic constraint logic (NCL): red (thin) edges have weight 1, blue (thick) edges have weight 2, and vertices have minimum inflow constraint of 2. (Left:) AND-vertex: edge C may be directed outward if and only if both edges A and B are directed inward. (Right:) OR-vertex: edge C may be directed outward if and only if either edge A or edge B is directed inward.

to prove PSPACE-hardness it suffices to construct AND- and OR-gadgets that simulate the behavior of the corresponding vertices and wiring capabilities.

3 Undirected Grid Graph Reachability, Revisited

When generalizing UGGR to UGGR with diagonal edges we obtain the following two results, depending on whether both or at most one diagonal edge within a subsquare of unity is allowed. For the former problem we obtain equivalence to UGGR under AC^0 reducibility.

Theorem 1. *UGGR with at most one diagonal edge within each subsquare of unity is equivalent to UGGR under* AC^0 *reducibility.*

Proof. Since ordinary UGGR is an instance of the more general problem it obviously reduces to the latter problem. Conversely, consider a UGGR instance G with at most one diagonal edge within each subsquare of unity. We construct an equivalent UGGR instance G'. Assume line-column coordinates for the vertices in the UGGR instance. For each vertex (i, j) in G we add to G' eight vertices $(3i, 3j+2)$, $(3i+1, 3j)$, $(3i+1, 3j+1)$, $(3i+1, 3j+2)$, $(3i+2, 3j+1)$, $(3i+2, 3j+2)$, $(3i+2, 3j+3)$, and $(3i+3, 3j+1)$ where the four vertices $(3i+1, 3j+1)$, $(3i+1, 3j+2)$, $(3i+2, 3j+1)$, and $(3i+2, 3j+2)$ from a square. Moreover, the four remaining vertices are connected to the square by the edges $((3i, 3j+2), (3i+1, 3j+2))$, $((3i+1, 3j), (3i+1, 3j+1))$, $((3i+2, 3j+2), (3i+2, 3j+3))$, and $((3i+2, 3j+1), (3i+3, 3j+1))$. Then a vertical edge $((i, j), (i+1, j))$ in G gives rise in G' to the two vertical edges $((3i+2, 3j+2), (3i+3, 3j+2))$ and $((3i+3, 3j+1), (3i+4, 3j+1))$. A horizontal edge $((i, j), (i, j+1))$ in G gives rise in G' to the two horizontal edges $((3i+1, 3j+2), (3i+1, 3j+3))$ and $((3i+2, 3j+3), (3i+2, 3j+4))$.

Finally, a diagonal edge $((i, j), (i+1, j+1))$ in G induces two edges $((3i+2, 3j+3), (3i+3, 3j+3))$ and $((3i+3, 3j+3), (3i+4, 3j+3))$ in G' with the new vertex $(3i+3, 3j+3)$. Similarly for a diagonal edge $((i, j+1), (i+1, j))$ in G the two edges $((3i+3, 3j+4), (3i+3, 3j+3))$ and $((3i+3, 3j+3), (3i+3, 3j+2))$ are added to G' with the new vertex $(3i+3, 3j+3)$.

The start and target vertices in G' are set accordingly. The reduction is illustrated in Figure 2. The correctness of the construction is easily seen by inspection. □

What happens if more than one diagonal is allowed within a unit square. In fact it is easy to see that all possible subgraphs that are unit squares with two diagonals that can occur in a undirected grid graph with diagonals, except the X-crossing, can be replaced by a unit square leading to a reachability equivalent undirected graph—see Figure 3. If X-crossings are allowed, we can prove equivalence to the more general UGR problem as shown next. This result is optimal by the previous consideration.

Theorem 2. *UGGR with diagonal edges is* AC^0 *equivalent to UGR. This result holds true even in case only X-crossings are allowed for diagonals.*

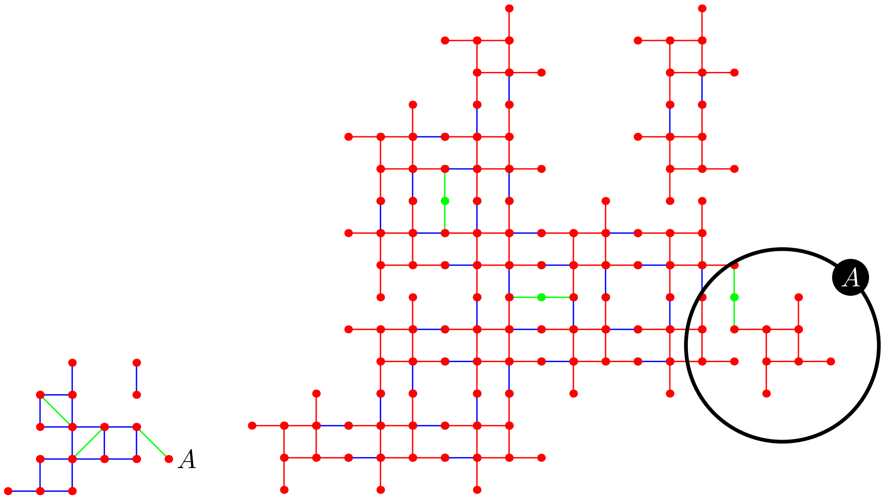

Fig. 2. (Left:) Undirected grid graph G with at most one diagonal edge within each subsquare of unity. The nodes are drawn red, the horizontal and vertical edges are blue, and the diagonal edges are shown green. Isolated vertices are not shown (Right:) Equivalent undirected grid graph G' obtained by construction form G. Here the red vertices and edges are induced by the vertices of the original graph G, the blue edges are induced by the horizontal and vertical edges of G, respectively, and the blue edges are induced by the diagonal vertices of G. The red subgraph within the circle named A is obtained by the reduction from the *single vertex* named A on the left.

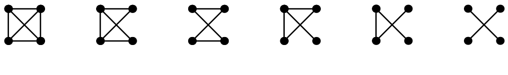

Fig. 3. All possible subgraphs that are unit squares with two diagonals up to mirroring and rotation, that can occur in a undirected grid graph with diagonals. All these subgraphs, except the last one (X-crossing), can be replaced by a unit square giving a reachability equivalent undirected graph.

Proof. Since a UGGR instance with diagonal edges is also a UGR instance, it remains to prove that UGR can be reduced to UGGR where the instance may have diagonal edges. Let (G, s, t) be an instance of the UGR problem, where $G = (V, E)$ is an undirected graph with vertices V and edges $E \subseteq V \times V$, and s and t are the source and the target vertices, respectively. Without loss of generality we may assume that $V = \{1, 2, \ldots, n\}$, that $s = 1$ and $t = n$, and that node n has a self-loop, i.e., edge (n, n) is in E. We reduce testing reachability in G to testing reachability in a undirected grid graph with diagonals constructed from four types of edge gadgets, namely *straight* edges, *X-crossing* edges, *up-route* edges, and *split-join* edges. See Figure 4 for a drawing of these four gadget types. The semantics of the X-crossing gadget on Figure 4 is that crossing edges do not touch. The constructed undirected grid graph will consist of a $2n \times O(n^4)$

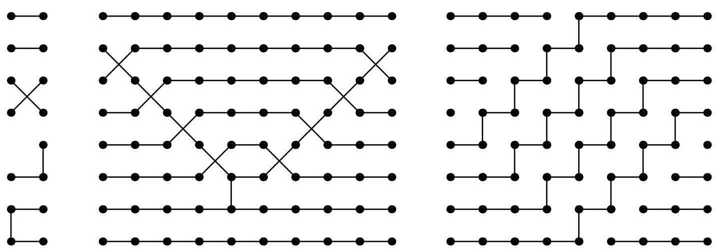

Fig. 4. (Left:) From top to bottom: straight edges, X-crossing edges, up-route edges (with dead ends), and split-join edges. (Middle:) Subgraph induced by an edge $(2, 3)$ of a directed graph G on four vertices. (Right:) Termination subgraph (slightly optimized in length to fit the line) after all edges of G were considered.

rectangular grid of nodes whose lines ("rows") are numbered $1, 2 \ldots, 2n$ from top to bottom. This grid is divided up into n identical blocks of size $2n \times m$, where $m \in O(n^3)$. The construction will maintain the property that a path of length k with $k > 0$ exists from node i to node j in G if and only if a path of length at most $m \cdot k$ exists from node $(i, 1)$ to node $(j, m \cdot k)$ in the rectangular grid.

A block is itself the concatenation from left to right of $O(n^2)$ edge layers, followed by a single termination layer. The edge layers are obtained from left to right by considering every edge in the graph G (in any order). The layer corresponding to edge (i, j) in G is constructed by first using a sequence of X-crossing gadgets to "bend line i downwards" across the lines below it. As the (bent) line i crosses the line $n + j - 1$, a split-join gadget is inserted to create a path from line i to line $n + j$. Using further X-crossing gadgets, line i is then bent back upwards and made to return to its original vertical position. The final (termination) layer in the block uses up-route gadgets to safely create paths from line $n + \ell$ to line ℓ, for $1 \leq \ell \leq n$. We illustrate the constructions of an edge layer and of a termination layer in Figure 4.

The upshot of concatenating n identical blocks is that node n is reachable from node 1 in G if and only if the rightmost node on line n is reachable from the leftmost node on line 1 in the layered graph. Clearly this construction can be done in AC^0. □

Thus we immediately obtain the following corollary by the L-completeness of UGR.

Corollary 3. *UGGR with diagonal edges is* L*-complete.* □

Finally, it is worth mentioning that Theorem 2 and Corollary 3 generalize to directed grid graph reachability (GGR) with *oriented* X-crossings and NL-completeness, since the subgraphs that are used in the proof of Theorem 2 and shown in Figure 4 can be oriented accordingly. This nicely contrasts the situation for ordinary GGR which is a special case of planar directed graph reachability which was recently proven to be contained in the class $\mathsf{UL} \cap \mathsf{coUL}$ in [2]—here UL is the set of problems that are accepted by unambiguous logspace bounded Turing machines. Thus, we can state the following corollary.

Corollary 4. *GGR with diagonal edges is* NL*-complete.* □

4 Xmas (Tree) Mazes

The name Xmas tree maze comes from a tree like shaped maze where one has to move a stick of length 2 to light up the star at the top of the Xmas tree—see `www.puzzlebeast.com/christmastree`. But one could also think of the tradition of putting a fir tree on top of a newly constructed roof framework. If the tree is very large and heavy, multiple persons are needed to carry the tree, while avoiding falling off the roof beams. An Xmas (tree) maze is played on an undirected grid graph with bended diagonals such that each subgraph that is a unit-square has at most one bended diagonal—here a bended diagonal is nothing other than a quarter of a circle. Thus, no crossings are allowed. The challenge is to bring a stick, whose length is a multiple of the unit, from a given starting position s to a target position t by a sequence of legal moves. A move is legal as long as all positions on the stick that are multiple of the unit stay in contact with the underlying undirected grid graph and end up at vertex positions. In case the stick is of unit length the additional property that both endpoints must end *simultaneously* on vertices applies. Hence, in this case it is not possible to move a stick around in a circle that is build by bended diagonals. An Xmas maze and its solution is shown in Figure 5. The undirected grid graph is drawn black and the stick of length 2 is red. Note that the stick has three positions marked by red dots that have to stay on the edges of the undirected grid graph through each legal movement.

4.1 Xmas Mazes with a Bounded Number of Sticks

After studying UGGR with diagonals in Section 3, we are now ready to investigate Xmas mazes with a constant number of sticks. First we show the following upper bound.

Theorem 5. *Deciding if an Xmas maze with a constant number of sticks has a solution can be done in deterministic logspace, regardless of the stick lengths.*

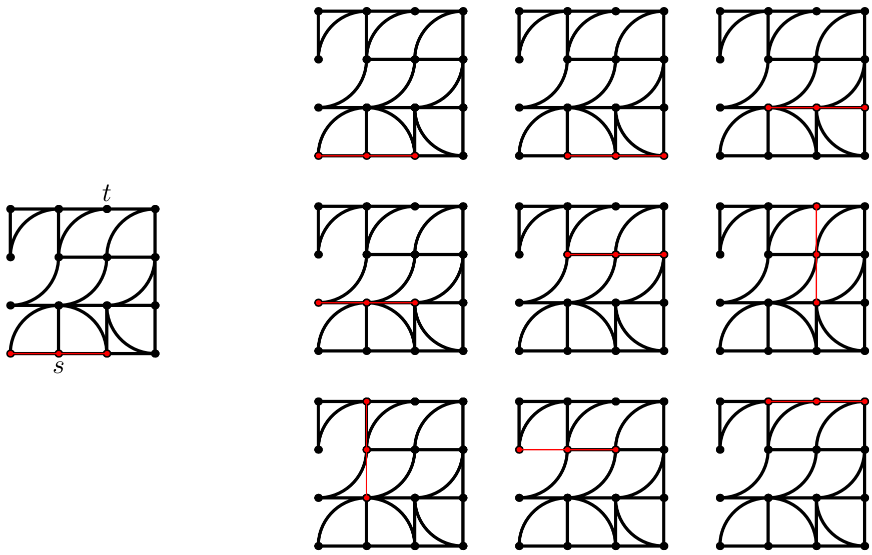

Fig. 5. Xmas maze of size 4×4 with a stick of length 2 (drawn in red with the three marked points that have to stay in contact with the underlying undirected grid graph) and its solution shown from left to right and top to bottom with the following movement of the stick: right, up, left, up-right, rotate clockwise, left, rotate counter clockwise, up-right. The source position s and the target position t refer to the middle position of the stick. This example is the first maze from `www.puzzlebeast.com/dryclean`.

Proof. For the containment in L it suffices to reduce Xmas maze solvability with a constant number of sticks to UGR. Given an Xmas maze instance, then it is easy to see that there are only a polynomial number of game configurations, since the number of sticks is bounded by a constant. These game configurations induce an undirected graph by the legal move relation, which is symmetric since a move can simply be undone. Thus, the Xmas maze can be solved by an appropriate UGR question induced by the source s and the target t. This proves containment within L by Reingold's result [6] on UGR. □

Next we prove some lower bounds on Xmas maze solvability. Obviously, an Xmas maze with a single stick of length 0 is AC^0 equivalent to UGGR. On the other hand, when allowing sticks of length 2 we already obtain L-completeness, even for a single stick.

Theorem 6. *Solving an Xmas maze with a constant number of sticks is* L*-complete, even with a single stick of length* 2*.*

Proof. The containment in L follows by Theorem 5. For the L-hardness we argue as follows. By Theorem 2 it suffices to reduce UGGR with X-crossings to the problem under consideration. Let G be an instance of the UGGR problem with X-crossings. We construct an Xmas maze G'. Assume line-column coordinates for the vertices in the UGGR instance. For each vertex (i, j) in G we add to G' the five vertices $(6i, 6j + 1)$, $(6i + 1, 6j)$, $(6i + 1, 6j + 1)$, $(6i + 1, 6j + 2)$, and $(6i + 2, 6j + 1)$ connected by a cross and the outer vertices $(6i, 6j + 1)$ and $(6i+1, 6j)$ are connected by an outward bended diagonal and also $(6i+1, 6j+2)$ and $(6i + 2, 6j + 1)$ are connected by a outward bended diagonal—see Figure 6.

Each non-diagonal edge in G gives rise to four edges of the same orientation

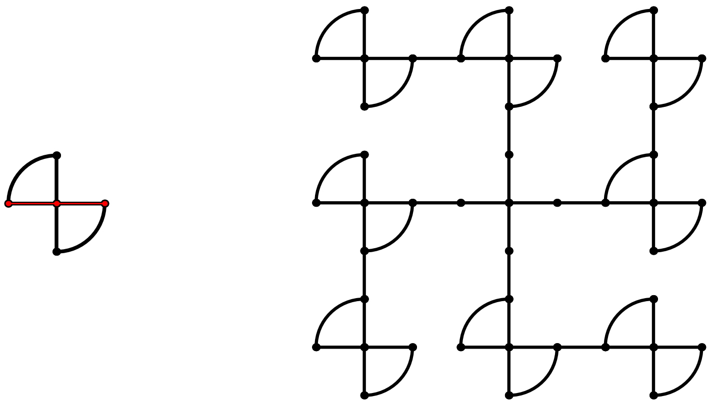

Fig. 6. (Left:) Subgraph of the underlying undirected grid graph of the Xmas maze induced by a single node in the UGGR plus X-crossing instance. The initial position of the stick of length 2 is also shown. (Right:) An X-crossing in G gives rise to the depicted subgraph that allows one to simulate an X-crossing with a single stick of length 1.

(horizontal or vertical). A vertical edge $((i,j),(i+1,j))$ in G gives rise in G' to the four vertical edges $((6i+2,6j+1),(6i+3,6j+1))$, $((6i+3,6j+1),(6i+4,6j+1))$, $((6i+4,6j+1),(6i+5,6j+1))$, and $((6i+5,6j+1),(6(i+1),6j+1))$. Moreover, the horizontal $((i,j),(i,j+1))$ in G gives rise in G' to the horizontal edges $((6i+1,6j+2),(6i+1,6j+3))$, $((6i+1,6j+3),(6i+1,6j+4))$, $((6i+1,6j+4),(6i+1,6j+5))$, and $((6i+1,6j+5),(6i+1,6(j+1)))$.

Finally, an X-crossing gives rise to the subgraph depicted in Figure 6—the four corner subgraphs correspond to the original vertices in G. The start and the target vertices in G' are set to $6 \cdot s + (1,1)$ and $6 \cdot t + (1,1)$ assuming component-wise multiplication. This completes the description of the reduction. Then it is easy to see that the original UGGR question is simulated by moving the stick from one corner subgraph to the next by rotating the stick accordingly to take the right direction. $\square$

It remains to consider the case when the sticks in an Xmas maze are all bounded in length by 1. When restricting to a single stick we obtain equivalence to UGGR under AC^0 reducibility. The case for a bounded number of sticks (larger than one) has to be left open, but by our previous considerations we deduce that this problem is hard for UGGR and is contained in L.

Theorem 7. *Solving an Xmas maze with a single stick of unit length is equivalent to UGGR under* AC^0 *reducibility.*

Proof. Let (G,s,t) be a UGGR instance. We construct an equivalent Xmas maze instance G' with a single stick of unit length. Assume line-column coordinates for the vertices in the UGGR instance. For each vertex (i,j) in G we add to G' four vertices $(2i,2j)$, $(2i,2j+1)$, $(2i+1,2j)$, $(2i+1,2j+1)$ and the four edges to form a square. Then a horizontal edge $((i,j),(i,j+1))$ in G gives rise in G' to the two horizontal edges $((2i+1,2j+1),(2i+1,2j+2))$ and

$((2i, 2j+1), (2i, 2j+2))$. A vertical edge $((i, j), (i+1, j))$ in G gives rise in G' to the two vertical edges $((2i+1, 2j), (2i+2, 2j))$ and $((2i+1, 2j+1), (2i+2, 2j+1))$. The start and target vertices in G' are set accordingly, i.e., $2 \cdot s$ and $2 \cdot t$, and the initial position of the stick connects $2 \cdot s$ with $2 \cdot s + (1, 0)$. Then it is easy to see that there is a path linking s to t in G if and only if the stick can move from its initial position to the target position in the Xmas maze G'.

Conversely we argue as follows: given an Xmas maze instance of size $n \times n$, first we construct a UGGR instance with diagonals. The idea for the reduction is that we only have to remember the possible positions of the single stick. Thus, since a complete $n \times n$ grid graph has $2 \cdot n \cdot (n-1)$ edges, the to be constructed UGGR instance G' with diagonals has exactly this number of vertices (naturally embedded into a grid)—each vertex lies at a position of an edge in the complete grid graph, representing the position of the stick on this very edge. The vertices of the undirected graph G' are induced by the movement of the single stick within the Xmas maze. To this end we consider all 4×3 subgraphs of the Xmas maze in more detail and assume that the stick of unit length resides in the middle in a horizontal position—see Figure 7. For a stick in a vertical position a similar argumentation will apply. The movement of the stick can be

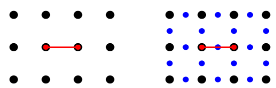

Fig. 7. (Left:) Xmas maze subgraph of size 4×3 (edges of the Xmas maze not shown) with a stick of unit length in the middle in horizontal position. (Right:) vertices of the UGGR instance G' with diagonals induced by the Xmas maze subgraph from the left. The vertices are drawn in blue.

as follows—in all cases see Figure 8:

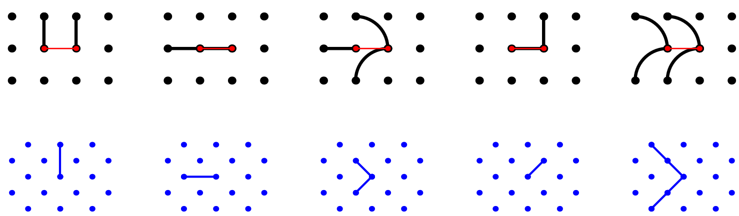

Fig. 8. Stick movements on a 4×3 Xmas maze subgraph that give rise to edges in the UGGR instance with diagonals. The shown Xmas mazes are examples for the movements only. (From left to right:) horizontal and vertical movements induce diagonal edges (first two pictures), rotation, special movement, and horizontal-to-vertical movement and *vice versa* give rise to unit square edges (next two pictures), and movement by two bended edges induce two unit length edges in sequence (last picture).

1. Horizontal and vertical stick movements induce diagonal edges in G'.
2. Rotation and special movements give rise to unit edges in G'. Also horizontal-to-vertical movement and *vice versa* does the same.

3. Movements by two bended diagonals in the Xmas maze would introduce long edges connecting vertices that have Manhattan distance two. These bended diagonals can also be used to rotate the stick either before or after the movement under consideration. Hence, the long edge is incident to at least one ordinary unit edge that is parallel. Therefore, we can safely introduce two consecutive unit length edges in G'.

This completes the description of the undirected grid graph G' with diagonals. Clearly this is an AC^0 reduction.

Next we show that G' doesn't contain an X-crossing. Then the equivalence to UGGR follows by our considerations in Subsection 3. For the sake of a contradiction assume that G' contains an X-crossing as subgraph. But then it is not hard to see that the original Xmas maze must have a unit square or a cross of unit edges which in turn gives rise to four additional edges that form a square with the diagonals connecting the opposite vertices in G'. This contradicts our assumption. Thus, G' is AC^0 equivalent to an UGGR instance (without diagonals). □

Note that the constructions in the proof of the previous theorem can be used to construct a UGGR instance from a given Xmas maze and then in turn an Xmas maze, now without bending diagonals, that is reachability equivalent to the original Xmas maze (if only a single stick of unit length is moved). Whether this is also possible for longer sticks is left open.

4.2 Xmas Mazes with an Unbounded Number of Sticks

Finally, we show that solving an Xmas maze is PSPACE-complete even for sticks of unit length if the number of sticks is not bounded. For proving PSPACE-hardness, we give a reduction from NCL.

Theorem 8. *Solving an Xmas maze with an unbounded number of sticks is* PSPACE*-complete, even if all stick are of unit size.*

Proof. Given an Xmas maze, a polynomial space bounded Turing machine can store the configuration and may simulate the sequence of movements of the sticks by simply guessing the sequence step by step. Since determinism and nondeterminism coincides for polynomial space by Savitch's theorem [7], the containment within PSPACE follows. It remains to show PSPACE-hardness.

We reduce NCL to the game under consideration. To this end it suffices to show how to simulate AND- and OR-vertices of constraint graphs as depicted in Figure 1 by Xmas maze subgames and to connect these subgames appropriately. The Xmas maze subgames simulating an AND- and an OR-vertex of NCL are depicted in Figure 9. By inspection it is not hard to see that in the left of Figure 9 (AND-vertex simulation) the both upper sticks at C may move inward, i.e., the corresponding edge is oriented outward, if and only if both sticks on the left at A and on the right at B are moved outward, that is, both corresponding edges are directed inward. Moreover, in the right of Figure 9 (OR-vertex simulation) the single edge at C may move inward, i.e., the corresponding edge is oriented

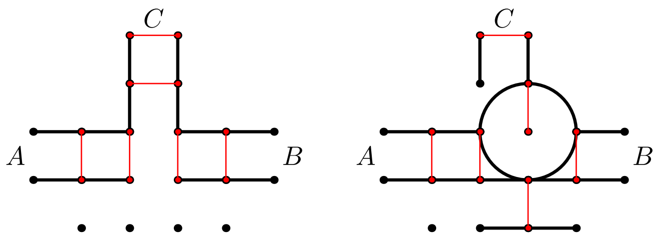

Fig. 9. (Left:) Xmas maze subgame simulating an AND-vertex. (Right:) Xmas maze subgame simulating an OR-vertex.

outward, if and only if both sticks on the left at A or the stick at B is moved outward, i.e., the corresponding edges are directed inward. The stick fixed in the middle of the circle enforces that at least one of the sticks at A (both sticks) and B are moved outward. Here is is worth mentioning that there is a particular movement that has to be discussed in detail—see Figure 10. The movement of

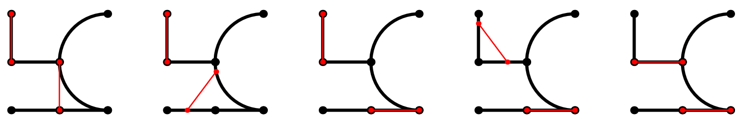

Fig. 10. Special movements of unit sticks.

the rightmost stick also appears in the Xmas maze subgame for the OR-vertex, while the latter movement of the leftmost stick has appeared in the previous section. The reader is invited to verify that the former movement doesn't change the functionality of the OR-vertex simulating Xmas maze subgame. This shows that the vertex simulation of NCL can be done with Xmas maze subgames.

Finally, it remains to show how to connect these simulating Xmas maze subgames appropriately. Moreover, the given construction also ensures that none of the sticks shown in Figure 9 can leave its submaze. The edge simulating Xmas maze subgame is drawn in Figure 11. There the two vertices at X (Y, respectively) are identified with the corresponding two vertices in the vertex simulating Xmas maze subgames at A, B, or C. The correctness of the construction is

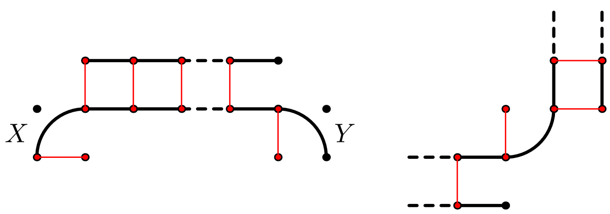

Fig. 11. (Left:) Xmas maze subgame simulating an edge connector. By construction none of the sticks (even in the vertex simulating Xmas maze subgames) can leave their devices. (Right:) subgame for bending an edge connector.

verified easily. This completes the proof of the PSPACE-completeness of solving

Xmas mazes with an unbounded number of sticks, even if all sticks are of unit size. □

5 Conclusions

We have investigated the computational complexity of reachability problems for grid graphs with diagonal edges, and of Xmas mazes. For grid graphs it turns out that allowing only one diagonal edge in each unit square does not increase the complexity, while allowing both diagonal edges makes the problem L-complete (NL-complete for directed grid graphs, respectively). For Xmas mazes we have seen that when the number of sticks in an Xmas maze is bounded by a constant, the solvability problems are related to the aforementioned grid graph reachability problems, depending on the stick lengths. If otherwise the number of sticks is unbounded, then the problem of solving such a maze becomes PSPACE-complete. Here, NCL turned out to be a great framework for proving PSPACE-hardness. Our findings on Xmas mazes are summarized in Table 1.

Stick length ℓ	Xmas maze variant: bounded (k sticks)	Xmas maze variant: unbounded
$\ell = 0$	$\cdot \equiv$ UGGR for $k = 1$	
	UGGR $\leq \cdot \in \mathsf{L}$	
$\ell = 1$	$\cdot \equiv$ UGGR for $k = 1$	PSPACE-complete
	UGGR $\leq \cdot \in \mathsf{L}$	
ℓ with $\ell \geq 2$	L-complete, even for $k = 1$	

Table 1. Xmas maze results for bounded and unbounded number of sticks in dependence of the stick length ℓ—here $\equiv$ and $\leq$ refer to AC^0 equivalence and AC^0 reducibility, respectively.

As the reader can see from the table, there are some open questions on Xmas mazes, in particular on the relation between complexity and stick length. What is the exact complexity of Xmas mazes with a constant number of sticks (more than one stick) all of length 1, or all of length 0? We only know that these problems are in L, and that UGGR can be reduced to both of them. What about the complexity if an unbounded number of sticks of length 0 can be used? Here we only have a PSPACE upper and a non matching lower bound of UGGR-hardness. This question is also related to similar problems for Alice mazes and rolling block puzzles—we refer to [4] for further results on these maze puzzle problems. In rolling block puzzles, the player has to roll blocks of size $\ell \times 1 \times 1$ through the corridors, and the goal is to move a designated block to a target position. The complexity of the special case, where all blocks are cubes of size $1 \times 1 \times 1$ is open, the best lower bound we know is UGGR hardness. In an Alice maze, tokens are moved according to arrows on the squares of the playing board, while possibly changing their individual speed on special squares. Again, the goal is to move a particular token to a target position. The complexity of solving Alice mazes with an unbounded number of tokens, which all have speed 1, and without any speed changing squares is open. Here, the best lower bound we know is NL-hardness. The relations to Xmas mazes are as follows. One can easily reduce the problem of solving rolling block puzzles with an unbounded number of cubes to the problem of solving an Xmas maze with an unbounded number of sticks of length 0. Also the latter problem can be reduced to the solvability problem for Alice mazes with an unbounded number of tokens, that all have speed 1, and without any speed changing squares. We believe these special cases of Xmas and rolling block mazes to be of equal complexity, while the mentioned Alice mazes could be computationally harder to solve. Whether this claim is actually true, must be verified by further research on the subject.

References

1. E. Allender, D. A. Mix Barrington, T. Chakraborty, S. Datta, and S. Roy. Planar and grid graph reachability problems. *Theory of Computing Systems*, 45(4):675–723, July 2009.
2. C. Bourke, R. Tewari, and N. V. Vinodchandran. Directed planar reachability is in unambiguous log-space. *ACM Transactions on Computation Theory*, 1(1), February 2009. Article 4.
3. E. D. Demaine and R. A. Hearn. A uniform framework for modeling computations as games. In *Proceedings of the 23th Annual IEEE Conference on Computational Complexity*, pages 149–162, College Park, Maryland, USA, June 2008. IEEE Computer Society Press.
4. M. Holzer and S. Jakobi. On the complexity of rolling block and Alice mazes. IFIG Research Report 1202, Institut für Informatik, Justus-Liebig-Universität Gießen, Arndtstr. 2, 35392 Gießen, Germany, March 2012.
5. C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
6. O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4):Article 17 (24 pages), September 2008.
7. W. J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, April 1970.

Institut für Informatik
Justus-Liebig-Universität Giessen
Arndtstr. 2, 35392 Giessen, Germany

Recent Reports

(Further reports are available at `www.informatik.uni-giessen.de`.)

H. Gruber, S. Gulan, *Simplifying Regular Expressions: A Quantitative Perspective*, Report 0904, August 2009.

M. Kutrib, A. Malcher, *Cellular Automata with Sparse Communication*, Report 0903, May 2009.

M. Holzer, A. Maletti, *An $n \log n$ Algorithm for Hyper-Minimizing States in a (Minimized) Deterministic Automaton*, Report 0902, April 2009.

H. Gruber, M. Holzer, *Tight Bounds on the Descriptional Complexity of Regular Expressions*, Report 0901, February 2009.

M. Holzer, M. Kutrib, and A. Malcher (Eds.), *18. Theorietag Automaten und Formale Sprachen*, Report 0801, September 2008.

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: Nondeterminism is Better than Determinism*, Report 0301, February 2003

M. Holzer, M. Kutrib, *Flip-Pushdown Automata: $k+1$ Pushdown Reversals are Better Than k*, Report 0206, November 2002

M. Holzer, M. Kutrib, *Nondeterministic Descriptional Complexity of Regular Languages*, Report 0205, September 2002

H. Bordihn, M. Holzer, M. Kutrib, *Economy of Description for Basic Constructions on Rational Transductions*, Report 0204, July 2002

M. Kutrib, J.-T. Löwe, *String Transformation for n-dimensional Image Compression*, Report 0203, May 2002

A. Klein, M. Kutrib, *Grammars with Scattered Nonterminals*, Report 0202, February 2002

A. Klein, M. Kutrib, *Self-Assembling Finite Automata*, Report 0201, January 2002

M. Holzer, M. Kutrib, *Unary Language Operations and its Nondeterministic State Complexity*, Report 0107, November 2001

A. Klein, M. Kutrib, *Fast One-Way Cellular Automata*, Report 0106, September 2001

M. Holzer, M. Kutrib, *Improving Raster Image Run-Length Encoding Using Data Order*, Report 0105, July 2001

M. Kutrib, *Refining Nondeterminism Below Linear-Time*, Report 0104, June 2001

M. Holzer, M. Kutrib, *State Complexity of Basic Operations on Nondeterministic Finite Automata*, Report 0103, April 2001

M. Kutrib, J.-T. Löwe, *Massively Parallel Fault Tolerant Computations on Syntactical Patterns*, Report 0102, March 2001

A. Klein, M. Kutrib, *A Time Hierarchy for Bounded One-Way Cellular Automata*, Report 0101, January 2001

M. Kutrib, *Below Linear-Time: Dimensions versus Time*, Report 0005, November 2000

M. Kutrib, *Efficient Universal Pushdown Cellular Automata and their Application to Complexity*, Report 0004, August 2000

M. Kutrib, J.-T. Löwe, *Massively Parallel Pattern Recognition with Link Failures*, Report 0003, June 2000

A. Klein, M. Kutrib, *Deterministic Turing Machines in the Range between Real-Time and Linear-Time*, Report 0002, February 2000

M. Kutrib, J.-T. Löwe, *Fault Tolerant Parallel Pattern Recognition*, Report 0001, February 2000

M. Kutrib, *Automata Arrays and Context-Free Languages*, Report 9907, October 1999

Th. Buchholz, A. Klein, M. Kutrib, *Iterative Arrays with Small Time Bounds*, Report 9906, August 1999