Satisfiability Parsimoniously Reduces to the Tantrix$^{\text{TM}}$ Rotation Puzzle Problem*

Dorothea Baumeister and Jörg Rothe

Institut für Informatik, Universität Düsseldorf, 40225 Düsseldorf, Germany

Abstract. Holzer and Holzer [HH04] proved that the Tantrix$^{\text{TM}}$ rotation puzzle problem is NP-complete. They also showed that for infinite rotation puzzles, this problem becomes undecidable. We study the counting version and the unique version of this problem. We prove that the satisfiability problem parsimoniously reduces to the Tantrix$^{\text{TM}}$ rotation puzzle problem. In particular, this reduction preserves the uniqueness of the solution, which implies that the unique Tantrix$^{\text{TM}}$ rotation puzzle problem is as hard as the unique satisfiability problem, and so is DP-complete under polynomial-time randomized reductions, where DP is the second level of the boolean hierarchy over NP.

Keywords: computational complexity, rotation puzzle, tiling of the plane, parsimonious reduction, counting problem.

1 Introduction

Tantrix$^{\text{TM}}$ is a puzzle game played with hexagonal tiles firmly arranged in the plane that each can be rotated around their axes. There are four different types of tiles (called *Sint*, *Brid*, *Chin*, and *Rond*, see Figure 1) that differ by the form of the three colored lines they each have, where colors are chosen among *red*, *yellow*, *blue*, and *green*. The objective of the game is to find a rotation of the given tiles so as to create long lines and loops of the same color. Since its invention in 1991 by Mike McManaway from New Zealand and its commercial launch, the Tantrix$^{\text{TM}}$ rotation puzzle has become extremely popular and commercially successful.

Holzer and Holzer [HH04] considered two variants of the Tantrix$^{\text{TM}}$ rotation puzzle problem, one with finitely many and one with infinitely many tiles in a given problem instance. They proved that the finite variant of this problem is NP-complete by reducing the NP-complete boolean circuit satisfiability problem (restricted to circuits with AND and NOT gates only) to it. They also showed that the infinite variant of the Tantrix$^{\text{TM}}$ rotation puzzle problem is undecidable, again employing a circuit construction. For other results on the complexity of problems related to Domino-like strategy games, we refer to Grädel [Grä90].

We consider two variants of the finite Tantrix$^{\text{TM}}$ rotation puzzle problem, its counting version and its unique version. The counting problem asks for the

* Supported in part by DFG grant RO 1202/9-3 and the Alexander von Humboldt Foundation's TransCoop program. URL: `http://ccc.cs.uni-duesseldorf.de/~rothe`

J. Durand-Lose and M. Margenstern (Eds.): MCU 2007, LNCS 4664, pp. 134–145, 2007.

number of solutions of a given rotation puzzle instance. The unique problem asks whether a given rotation puzzle instance has exactly one solution. Our main result is that the satisfiability problem parsimoniously reduces to the Tantrix$^{\text{TM}}$ rotation puzzle problem.

The class #P was introduced by Valiant [Val79] to capture the complexity of counting the solutions of NP problems. Parsimonious reductions between NP counting problems—such as ours—preserve the precise number of solutions. This is an important property for at least two reasons. First, the structure of the solution space is preserved by a parsimonious reduction from A to B, since solutions of A are mapped bijectively to solutions of B in polynomial time. Second, parsimonious reductions can be used to prove lower bounds for the unique versions of NP problems. In particular, we apply our above-mentioned parsimonious reduction to prove that the unique Tantrix$^{\text{TM}}$ rotation puzzle problem is DP-complete[1] under polynomial-time randomized reductions in the sense of Valiant and Vazirani [VV86].

While many standard reductions between NP-complete problems are easily seen to be parsimonious, there are a number of exceptions. For example, Barbanchon [Bar04] showed that the (planar) satisfiability problem is parsimoniously polynomial-time reducible to the (planar) 3-colorability problem via a rather sophisticated construction. Other examples of nontrivial parsimonious reductions can be found in [Pap94]. Holzer and Holzer's reduction, however, is not parsimonious [HH04]. The main purpose of this paper is to show how to modify their reduction so as to make it parsimonious.

2 Preliminaries

2.1 Definition of Some Complexity-Theoretic Notions

Fix the alphabet $\Sigma = \{0, 1\}$, and let Σ^* denote the set of strings over Σ. As is common, decision problems are suitably encoded as languages over Σ. For any language $A \subseteq \Sigma^*$, let $\|A\|$ denote the number of elements in A. For some background on computational complexity theory, we refer to any standard textbook of this field, e.g., [Pap94, Rot05]. Let NP denote the class of problems solvable in nondeterministic polynomial time. Generalizing NP, Papadimitriou and Yannakakis [PY84] introduced the class $\text{DP} = \{A - B \mid A, B \in \text{NP}\}$ to capture the complexity of NP-hard or coNP-hard problems that seemingly are neither in NP nor in coNP. In particular, they showed that DP contains a number of *uniqueness problems*, *critical graph problems*, and *exact optimization problems*, and they showed some of these problems complete for DP; see also the recent survey [RR06]. Note that DP was later generalized by Cai et al. [CGH$^+$88, CGH$^+$89], who introduced the boolean hierarchy over NP. Note that DP is the second level of this hierarchy.

[1] DP is the set of differences of any two NP sets [PY84]; so $\text{NP} \subseteq \text{DP}$, and it is considered most unlikely that both classes are equal.

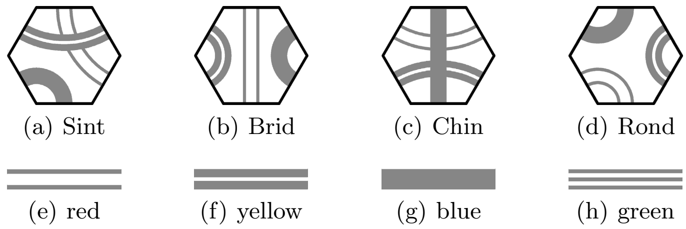


Fig. 1. Tantrix$^{\text{TM}}$ tiles and colors

In his seminal paper, Valiant [Val79] initiated the study of counting problems and introduced the important counting class #P. Members of #P are referred to as NP *counting problems*. A well-known NP counting problem is #SAT, the counting version of the satisfiability problem: Given a boolean formula, how many satisfying assignments does it have?

Definition 1 (Valiant [Val79]). *Let NPTM be a shorthand for nondeterministic polynomial-time Turing machine. For any NPTM M and any input x, let $acc_M(x)$ denote the number of accepting computation paths of $M(x)$, i.e., acc_M is a function mapping from Σ^* to $\mathbb{N}$. Define the function class* $\#\mathrm{P} = \{acc_M \mid M \text{ is an NPTM}\}$.

We now define the notion of *(polynomial-time) parsimonious reducibility*, which will be used to compare the hardness of solving NP counting problems. Intuitively, an NP counting problem f parsimoniously reduces to an NP counting problem g if the instances of f can be transformed into instances of g such that the number of solutions of f are preserved under this transformation.

Definition 2. *Let f and g be any two given counting problems mapping from Σ^* to $\mathbb{N}$. We say f* (polynomial-time) parsimoniously reduces to *g (denoted by $f \leq_{par}^{p} g$) if there exists a polynomial-time computable function ρ such that for each $x \in \Sigma^*$, $f(x) = g(\rho(x))$. If F and G are the* NP *decision problems corresponding to the* NP *counting problems f and g with $f \leq_{par}^{p} g$, we will also say that F parsimoniously reduces to G.*

2.2 Variants of the Tantrix Rotation Puzzle Problem

The Tantrix$^{\text{TM}}$ rotation puzzle has four kinds of hexagonal tiles—the *Sint*, the *Brid*, the *Chin*, and the *Rond*—each of which has three colored lines, where the colors are chosen among *red*, *yellow*, *blue*, and *green*, see Figure 1(a)–(d). This gives a total of 56 different tiles. Since we aren't using actually colored figures, we encode the colors as shown in Figure 1(e)–(h).

Holzer and Holzer [HH04] showed that the decision problem Tantrix$^{\text{TM}}$ rotation puzzle (which we denote by TRP, for short) is NP-complete. In this paper, we introduce and study #TRP, the counting version of TRP.

We now briefly describe the formalism introduced by Holzer and Holzer [HH04] to define TRP, since the same formalism is useful for defining #TRP. In

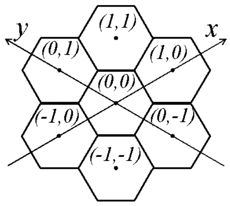


Fig. 2. A two-dimensional hexagonal coordinate system

particular, to represent the instances of both these problems, a two-dimensional hexagonal coordinate system is used, see Figure 2. In this system, two distinct pairs $a = (u, w)$ and $b = (v, x)$ from $\mathbb{Z}^2$ are adjacent if and only if one of the following four conditions is satisfied:

1. $u = v$ and $|w - x| = 1$,
2. $|u - v| = 1$ and $w = x$,
3. $u - v = 1$ and $w - x = 1$, and
4. $u - v = -1$ and $w - x = -1$.

Let T be the set of all Tantrix[TM] tiles. Let $\mathcal{A}$ be a (partial) function mapping the elements of $\mathbb{Z}^2$ to T, i.e., for those $v \in \mathbb{Z}^2$ on which $\mathcal{A}$ is defined, $\mathcal{A}(v)$ is the type of the tile located at v. The set $\mathit{shape}(\mathcal{A}) = \{v \in \mathbb{Z}^2 \mid \mathcal{A}(v) \text{ is defined}\}$ gives the positions in $\mathbb{Z}^2$ at which tiles are placed. For all $a, b \in \mathit{shape}(\mathcal{A})$, $\mathcal{A}(a)$ is adjacent to $\mathcal{A}(b)$ if and only if a is adjacent to b.

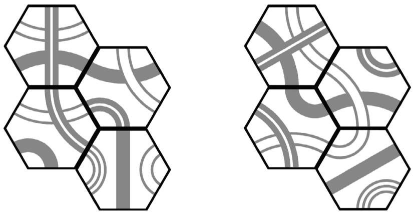
TRP is then defined as follows (note that the initial orientation is not specified, as it doesn't matter for the question of whether the decision problem TRP is solvable) [HH04]:[2]

Name: Tantrix[TM] Rotation Puzzle (TRP, for short).
Given: A finite shape function $\mathcal{A} : \mathbb{Z}^2 \to T$, appropriately encoded as a string.
Question: Is the rotation puzzle defined by $\mathcal{A}$ solvable, i.e., does there exist a rotation of the given tiles at their positions such that at each joint edge of two adjacent tiles the corresponding colors match?

For any given TRP instance $\mathcal{A}$, a *solution of* $\mathcal{A}$ is a specification (in some appropriate encoding) of each tile in $\mathit{shape}(\mathcal{A})$ in some particular orientation such that for each joint edge of two adjacent tiles the corresponding colors match. Figure 3 gives an example of a rotation puzzle instance and its solution. Let $\textsc{Sol}_{\mathrm{TRP}}(\mathcal{A})$ denote the set of solutions of a given TRP instance $\mathcal{A}$. So $\mathcal{A}$ is in TRP (viewed as a language) if and only if the set $\textsc{Sol}_{\mathrm{TRP}}(\mathcal{A})$ is nonempty.

We now define the counting version and the unique version of TRP, which will be considered in Sections 3 and 4.

[2] As noted by Holzer and Holzer [HH04], there is a difference between their definition of TRP, which allows holes in TRP instances, and the original Tantrix[TM] game, which does not allow holes. The problem of whether the analog of TRP *without* holes still is NP-complete is open.



(a) Puzzle (b) Solution

Fig. 3. An example of a TRP instance and its solution

Definition 3. *1. The* Tantrix$^{\text{TM}}$ rotation puzzle counting problem *is the function* $\#TRP : \Sigma^* \to \mathbb{N}$ *defined by*

$$\#TRP(\mathcal{A}) = \|\text{SOL}_{TRP}(\mathcal{A})\|,$$

where we assume that inputs $\mathcal{A}$ *are appropriately encoded as strings in* Σ^* *and function values are nonnegative integers (represented in binary).*

2. The unique Tantrix$^{\text{TM}}$ rotation puzzle problem *is defined by*

$$\textit{Unique-TRP} = \{\mathcal{A} \mid \#TRP(\mathcal{A}) = 1\}.$$

3 Satisfiability Parsimoniously Reduces to the Tantrix$^{\text{TM}}$ Rotation Puzzle Problem

In this section, we prove our main result:

Theorem 4. $\#SAT \leq^p_{par} \#TRP$.

The proof of Theorem 4 will be presented in Sections 3.1, 3.2, 3.3, and 3.4.

To prove TRP NP-complete, Holzer and Holzer [HH04] gave a reduction from the NP-complete problem Circuit$_{\wedge,\neg}$-SAT (see Cook [Coo71]): Given a boolean circuit C with AND and NOT gates, does there exist a truth assignment to the input gates of C such that C under this assignment evaluates to *true*? Holzer and Holzer's construction simulates the computation of such a boolean circuit C by a Tantrix$^{\text{TM}}$ rotation puzzle such that C evaluates to *true* for some assignment to its variables if and only if the puzzle has a solution.

Our construction will modify Holzer and Holzer's reduction [HH04] in such a way that there is a one-to-one correspondence between the solutions of the given Circuit$_{\wedge,\neg}$-SAT instance and the solutions of the resulting rotation puzzle instance; hence our reduction is parsimonious. The reduction employs planar cross-over gates (consisting of AND and NOT gates only) to avoid wire crossings of the given circuit; for technical details and examples, see [HH04].

To simulate the circuit by a rotation puzzle, a number of subpuzzles are used. The color blue in these subpuzzles will represent the truth value *true*, and the color red will represent *false*. This color encoding at the inputs and outputs of the subpuzzles thus represent the truth values of the circuit's gates and wires.

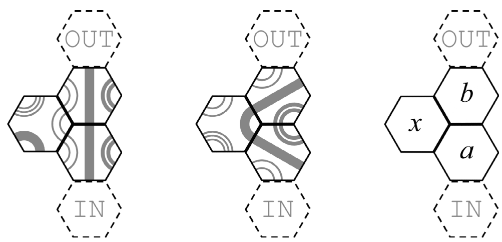
Due to space limitations, the original subpuzzles from [HH04] that are given in the full version of this paper [BR07] to allow comparison, are omitted here. Our arguments about the original subpuzzles refer to the notation used in [BR07].

3.1 Wire Subpuzzles

Wires of the circuit are simulated by the subpuzzles WIRE, MOVE, and COPY.

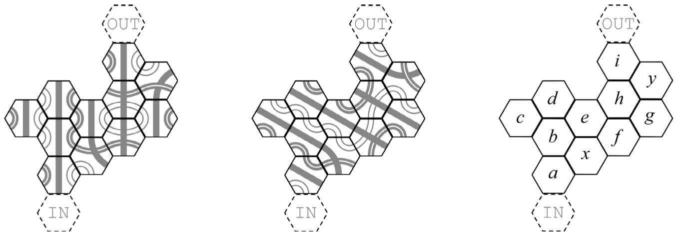
Figure 4 shows the modified WIRE subpuzzle, which simply represents a vertical wire. Longer wires can be built by using several WIRE subpuzzles. A single WIRE has height two, which implies that all other subpuzzles must have even height. (Otherwise it wouldn't be possible to simulate a circuit by a rotation puzzle.) It is easy to see that the original WIRE subpuzzle from [HH04] (see also [BR07, Figure 11]) has more than one valid solution with the input colors blue and red. In particular, tile a and tile b have two possible orientations for each input color, so there are four possible solutions. However, by inserting a *Rond* in the colors blue, red, and green at position x, we obtain a unique solution. If the input color is blue, there is a blue vertical line. Tiles a and b now must have red at the edge adjacent to tile x, since x doesn't have yellow. If the input color is red, tiles a and b have a choice between either blue or yellow for the edge joint with x. Again, since x doesn't have yellow, the solution is unique.

Figure 5 presents the modified MOVE subpuzzle by which a wire can be moved by two positions to the left or to the right. Consider a move to the right (a move to the left can be handled analogously). The original MOVE subpuzzle from [HH04] (see also [BR07, Figure 12]) has again more than one valid solution. To eliminate this ambiguity, we do the following. Suppose the input color is blue. Since red and yellow are symmetric in all tiles of the original MOVE subpuzzle (see [BR07, Figure 12]), there are two symmetric solutions. Also, tiles a and i have two possible orientations for each such solution. However, inserting a *Sint* in the colors red, yellow, and blue at position x enforces that the tile b has yellow at the edge adjacent to x, since e has blue at the edge adjacent to x for both possibilities. Fixing the orientation of tile b also fixes the orientation of all other tiles except a and i. Moreover, a's orientation is also uniquely determined by the tile at position x. First, for the joint edge of tiles a and x, one can choose among the colors red and yellow, but since yellow already must be the color x uses for



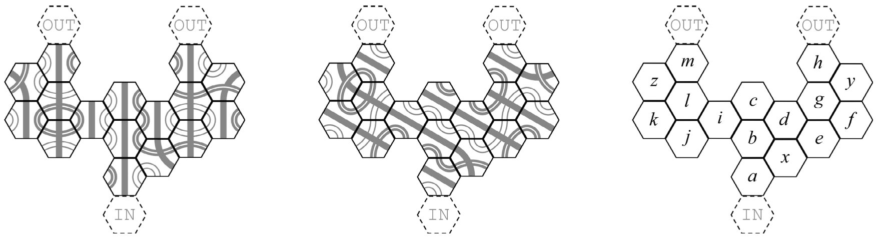
(a) In: *true* (b) In: *false* (c) Scheme

Fig. 4. Modified subpuzzle WIRE



(a) In: *true* (b) In: *false* (c) Scheme

Fig. 5. Modified subpuzzle MOVE



(a) In: *true* (b) In: *false* (c) Scheme

Fig. 6. Modified subpuzzle COPY

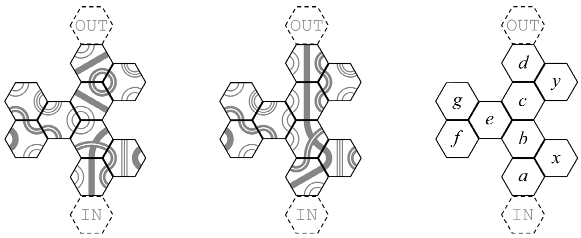
the edge joint with b, x must have red as the color for the edge joint with a. Second, to uniquely determine the orientation of tile i, insert another *Sint* in the colors red, yellow, and blue at position y. Since tile h has color yellow at the edge joint with y, symmetric arguments as above work for fixing the orientation of i via y. The case of red being the input color can be handled similarly.

Finally, Figure 6 presents the modified COPY subpuzzle, which can be used to “split” a wire into two copies. Its structure is akin to the MOVE subpuzzle, though it is wider and has two outputs. Due to symmetry, the original COPY subpuzzle from [HH04] (see also [BR07, Figure 13]) has again more than one valid solution. However, we can enforce a unique solution (for both input colors, blue and red) by inserting three asymmetric *Sint* tiles in the colors red, blue, and yellow at positions x, y, and z. The argument then is similar to the one for the MOVE subpuzzle.

3.2 Gate Subpuzzles

In order to simulate a boolean circuit with AND and NOT gates, we need the subpuzzles AND and NOT corresponding to these gates.

Figure 7 presents the modified NOT subpuzzle, which negates the input value by flipping the colors blue and red. In the original NOT subpuzzle from [HH04] (see also [BR07, Figure 14]), there is only one possible orientation for the tiles e, f, and g, since the tiles c, b, and f do not contain green. Thus tiles c and b must have red at the edge adjacent to tile e. It follows that for each input

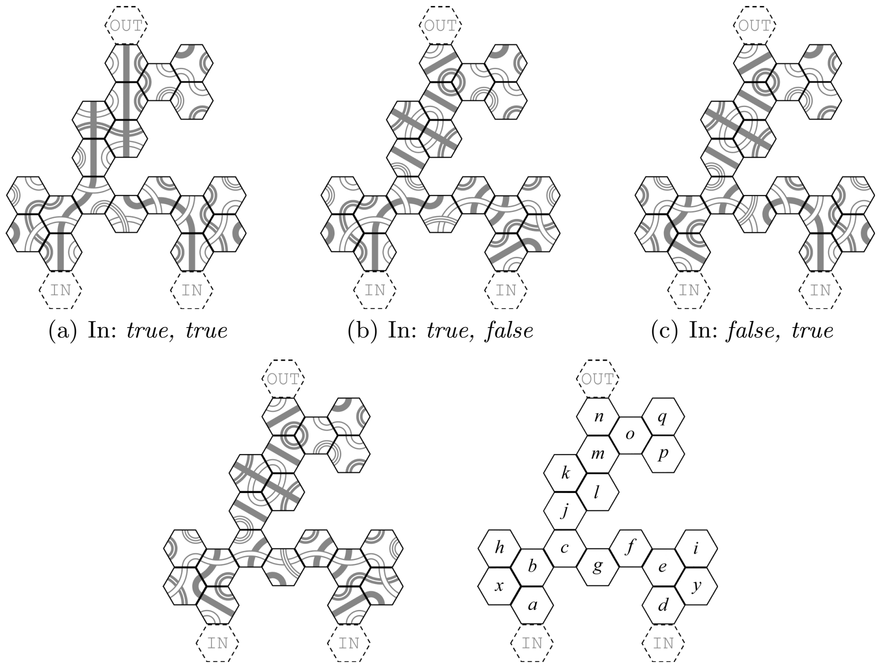


(a) In: *true* (b) In: *false* (c) Scheme

Fig. 7. Modified subpuzzle NOT

color only two orientations are possible for tiles a and d. Inserting a *Brid* in the colors blue, yellow, and green at position x uniquely determines the orientation of tile a. Since x does not contain red, we have that tile a is forced to choose yellow at the edge adjacent to x if the input color was blue. On the other hand, if the input color was red, a has a choice between blue and green for this color because x has blue at the edge joint with b. However, since a doesn't contain green, this uniquely determines the orientation of a. The orientation of tile d can be made unique by inserting a *Rond* in the colors yellow, red, and green at position y. For both input colors, c has yellow at the edge joint with y, so d and y can share either yellow or green. Since tile d contains no green, its orientation is uniquely determined. Thus, for both input colors, a valid solution of the NOT subpuzzle is uniquely determined.

Figure 8 presents the somewhat more complicated modified AND subpuzzles. Similarly as for the original NOT subpuzzle, in the original AND subpuzzle from [HH04] (see also [BR07, Figure 15]) the tiles o, p, and q have only one possible orientation, due to the colors at the joint edges with tiles m and n. The output of these subpuzzles is determined by the orientation of tile c. If c's color at the edge joint with j is blue, then the output color also is blue. This is the case exactly if both inputs have color blue. In all other cases, the color at the joint edge of c and j is yellow, which implies that the output color will be red. For each of these subpuzzles' upper part, the orientations are determined by the color at the joint edge of c and j. For their lower parts, however, several solutions are possible. Again, there are two possible orientations of the tiles a and d. Since the tiles h and i have only one connection to the remaining subpuzzle each, they too have two possible orientations, independent of the input color. However, since the tiles h and i do not contain blue, the input color uniquely determines the orientation of tiles b and e. The tiles c, g, and f have a different orientation for each combination of input colors, which implies that the correct output color is passed on by tile j to the upper part of the subpuzzle. However, two orientations are possible for tile g, which is a *Rond*. Due to its symmetry and since its neighbors are c and f, it cannot receive a unique orientation by the colors at the joint edges with these neighbors in the original subpuzzle (see [BR07, Figure 15]).



(a) In: *true, true* (b) In: *true, false* (c) In: *false, true*

(d) In: *false, false* (e) Scheme

Fig. 8. Modified subpuzzle AND

The orientations of tiles a and h (respectively, of tiles d and i) can be made unique by inserting a *Sint* in the colors blue, yellow, and red at position x (respectively, at position y). The orientation of tile g cannot be made unique by inserting another tile. For these subpuzzles, there exist four possible combinations of input colors, and because of the given colors at joint edges it is not possible to insert a new tile adjacent to g. This implies that we need to replace existing tiles by different tiles. So, to obtain a unique solution, we replace the color yellow by green in the tiles j and c, and tile g is replaced by a *Sint* in the colors green, blue, and red. Replacing yellow by green in j and c is easily possible. Tile j has yellow at the edge joint with c only in some cases, and c's yellow is also replaced by green. Also, tile c has yellow at the edge adjacent to g, which now is green at this edge. By the new tile at position g, the orientation of f changes if the right input color is blue, since yellow is no longer possible as the color for the joint edge. These replacements do not alter the behavior of the AND subpuzzles, other than giving the desired effect that solutions now are unique.

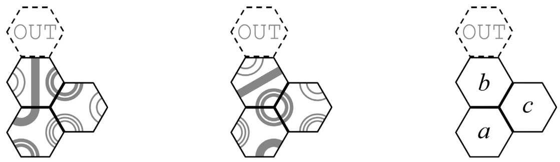
The shapes of the single subpuzzles have changed by these replacements, which might imply unintended interactions between various subpuzzles. However, essentially by the argument given by Holzer and Holzer [HH04] about the minimal horizontal distance between any two wires and/or gates being at least four, such undesired interactions do not occur.

3.3 Input and Output Subpuzzles

The BOOL subpuzzle represents the input gates of the circuit. This subpuzzle has only two valid solutions, either its output is blue (if the corresponding input variable is *true*), or it is red (if the corresponding input variable is *false*). This ensures that subsequent subpuzzles can obtain only these two colors as input.

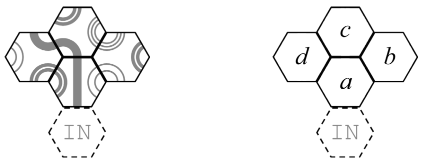
The subpuzzle TEST tests whether the function value computed by the circuit is *true* or not. This subpuzzle has only one valid solution, namely that its input is blue (which means that the circuit evaluates to *true*).

Obviously, neither of these subpuzzles, BOOL and TEST, do require any modification, and they are the only subpuzzles from [HH04] not modified. For completeness, we present them in Figures 9 and 10.



(a) Out: *true* (b) Out: *false* (c) Scheme

Fig. 9. Subpuzzle BOOL, see [HH04]



(a) Output (b) Scheme

Fig. 10. Subpuzzle TEST, see [HH04]

3.4 Proof of Theorem 4

We are now ready to prove Theorem 4. Let SAT denote the satisfiability problem.

Lemma 5. *SAT parsimoniously reduces to Circuit*$_{\wedge,\neg}$*-SAT.*

Proof. Note that the problems SAT and Circuit-SAT (which is the same as Circuit$_{\wedge,\neg}$-SAT except with OR gates allowed as well) are equivalent under parsimonious reductions [Pap94]. Since OR gates can be expressed by AND and NOT gates without changing the number of solutions, this gives a parsimonious reduction from SAT to Circuit$_{\wedge,\neg}$-SAT. ❑

Now, the parsimonious reduction from SAT to TRP immediately follows from Lemma 5 and the construction and the arguments presented in Sections 3.1, 3.2, and 3.3.

4 The Unique Tantrix$^{\text{TM}}$ Rotation Puzzle Problem Is DP-Complete Under Randomized Reductions

Valiant and Vazirani introduced *randomized polynomial-time reductions* in their work showing that NP is as easy as detecting unique solutions [VV86]. We will use $\leq_{ran}^{p}$ to denote their type of reductions. In particular, Valiant and Vazirani [VV86] proved that Unique-SAT, the unique version of SAT, is $\leq_{ran}^{p}$-complete in DP (see also Chang, Kadin, and Rohatgi [CKR95]).

Theorem 6. *1. Unique-SAT parsimoniously reduces to Unique-TRP.*
2. Unique-TRP is DP*-complete under* $\leq_{ran}^{p}$*-reductions.*

Proof. To prove the first part, note that by Lemma 5 and Theorem 4, we obtain a parsimonious reduction from SAT to TRP. It follows that Unique-SAT parsimoniously reduces to Unique-TRP.

The second part follows from the first part and Valiant and Vazirani's above-mentioned result that Unique-SAT is $\leq_{ran}^{p}$-complete in DP, and from the obvious fact that Unique-TRP is in DP. ❑

Acknowledgments. We thank the anonymous MCU 2007 referees for their helpful comments.

References

[Bar04] Barbanchon, R.: On unique graph 3-colorability and parsimonious reductions in the plane. Theoretical Computer Science 319(1–3), 455–482 (2004)

[BR07] Baumeister, D., Rothe, J.: Satisfiability parsimoniously reduces to the Tantrix$^{\text{TM}}$ rotation puzzle problem. Technical Report cs.CC/0705.0915, ACM Computing Research Repository (CoRR) (May 2007)

[CGH$^+$88] Cai, J., Gundermann, T., Hartmanis, J., Hemachandra, L., Sewelson, V., Wagner, K., Wechsung, G.: The boolean hierarchy I: Structural properties. SIAM Journal on Computing 17(6), 1232–1252 (1988)

[CGH$^+$89] Cai, J., Gundermann, T., Hartmanis, J., Hemachandra, L., Sewelson, V., Wagner, K., Wechsung, G.: The boolean hierarchy II: Applications. SIAM Journal on Computing 18(1), 95–111 (1989)

[CKR95] Chang, R., Kadin, J., Rohatgi, P.: On unique satisfiability and the threshold behavior of randomized reductions. Journal of Computer and System Sciences 50(3), 359–373 (1995)

[Coo71] Cook, S.: The complexity of theorem-proving procedures. In: Proceedings of the 3rd ACM Symposium on Theory of Computing, pp. 151–158. ACM Press, New York (1971)

[Grä90] Grädel, E.: Domino games and complexity. SIAM Journal on Computing 19(5), 787–804 (1990)

[HH04] Holzer, M., Holzer, W.: Tantrix$^{\text{TM}}$ rotation puzzles are intractable. Discrete Applied Mathematics 144(3), 345–358 (2004)

[Pap94] Papadimitriou, C.: Computational Complexity. Addison-Wesley, Reading (1994)

[PY84] Papadimitriou, C., Yannakakis, M.: The complexity of facets (and some facets of complexity). Journal of Computer and System Sciences 28(2), 244–259 (1984)

[Rot05] Rothe, J.: Complexity Theory and Cryptology. An Introduction to Cryptocomplexity. EATCS Texts in Theoretical Computer Science. Springer, Heidelberg (2005)

[RR06] Riege, T., Rothe, J.: Completeness in the boolean hierarchy: Exact-Four-Colorability, minimal graph uncolorability, and exact domatic number problems – a survey. Journal of Universal Computer Science 12(5), 551–578 (2006)

[Val79] Valiant, L.: The complexity of computing the permanent. Theoretical Computer Science 8(2), 189–201 (1979)

[VV86] Valiant, L., Vazirani, V.: NP is as easy as detecting unique solutions. Theoretical Computer Science 47, 85–93 (1986)