“Planar” tautologies hard for Resolution

Stefan Dantchev[1,2]
[1]BRICS[*], Dept. of Computer Science
University of Aarhus
540 Ny Munkegade
DK-8000 Aarhus C, Denmark
dantchev@dcs.qmw.ac.uk

Søren Riis[2]
[2]Dept. of Computer Science
Queen Mary, University of London
Mile End Road
London E1 4NS, UK
smriis@dcs.qmw.ac.uk

Abstract

We prove exponential lower bounds on the resolution proofs of some tautologies, based on rectangular grid graphs. More specifically, we show a $2^{\Omega(n)}$ lower bound for any ***resolution proof*** *of* ***the mutilated chessboard problem*** *on a $2n \times 2n$ chessboard as well as for* ***the Tseitin tautology*** *based on the $n \times n$* ***rectangular grid graph****. The former result answers a 35 year old conjecture by McCarthy.*

1 Introduction

In the paper, we prove an exponential lower bound for any resolution proof of the mutilated chessboard problem as well as for the Tseitin tautologies on a rectangular grid graph.

Exponential lower bounds for resolution are known for matching problems based on the complete bipartite graph $K_{n+1,n}$ as well as for a special class of graphs, namely expanders (see [4], [8], [3]). Exponential lower bounds for Tseitin tautologies are also known for expander graphs only [7]. In the recent paper [2], a common framework is given that generalises and simplifies all the known proofs. Unfortunately, it does not work for tautologies based on planar graphs.

Thus our main contribution is that we obtain exponential lower bounds for tautologies, based on grid graphs. The main tool, we use in our proofs, is the representation of resolution proofs as Prover-Adversary games. It is introduced by Pudlak in his recent paper [6]. On a technical level, our contribution is a new way to introduce randomness in Adversary's strategy (although Pudlak, himself, speaks about “super-strategy” rather than “randomised strategy”). In doing so, we introduce the concept of tiling games. It turns out that the combination of our reduction of the original problems to tiling games and Pudlak's idea of considering proofs as games gives very “clean” proofs of the lower bounds.

The paper is organised as follows. First, we define the two problems and explain briefly Pudlak's idea of considering resolution proofs as games. We then introduce tiling games and prove lower bounds for them. Finally, we show the reduction from the original problems to tiling games.

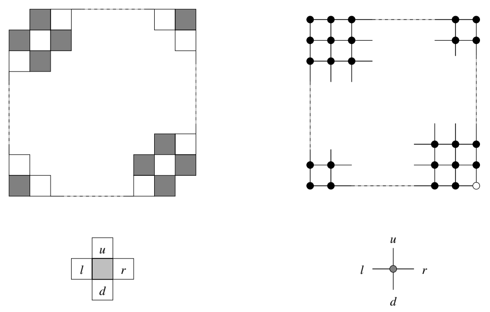

Figure 1. The two original problems

Mutilated chessboard This problem has the distinction to be the earliest proposed hard problem for theorem provers [5]. The problem is the following: given a $2n \times 2n$ chessboard with two diagonally opposite squares missing (see the left side of Fig. 1), prove that it cannot be covered with dominoes. We can consider it as a matching problem (the left part of Fig. 2): squares are vertices of the graph, and there is an edge between every two neighbouring squares. Thus one component of the bipartite graph consists of black squares and the other consists of white ones. Two missing squares are of the same colour which implies one of the components in the graph has two more vertices than the other. That is why there is no perfect matching, i.e., dominoes tiling of the mutilated chessboard.

The formalisation of the problem as a set of clauses is as follows. For every square, we introduce (at most) four

[*]**B**asic **R**esearch **I**n **C**omputer **S**cience, Centre of the Danish National Research Foundation

variables u, r, d, l corresponding to the four possible ways of covering a square by a domino. We then write down the following clauses, saying that every square is covered exactly once:

1. $\{u, r, d, l\}$
2. $\{\overline{u}, \overline{r}\}$, $\{\overline{u}, \overline{d}\}$, $\{\overline{u}, \overline{l}\}$, $\{\overline{r}, \overline{d}\}$, $\{\overline{r}, \overline{l}\}$, $\{\overline{d}, \overline{l}\}$

Whenever a variable does not make sense, i.e., a domino, going outside the chessboard, we replace the corresponding variable by "false".

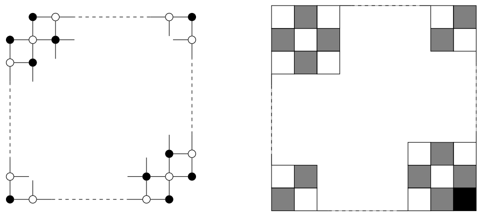

Figure 2. The "reverse" formulations

Tseitin tautologies on grid graphs The definition of the problem is as follows. Given a undirected graph, we attach a propositional variable to every edge. We also select one vertex and label it by "true", all others are labelled by "false". We require the exclusive-or of all the adjacent edges of every vertex to be equal to its label. Obviously, this is impossible as every variable occurs exactly twice in the exclusive-or part of these equations, but the exclusive-or of all the labels is "true".

On a $n \times n$ rectangular grid graph, we colour white one of the corners, and all the other vertices are black (see the right side of Fig. 1). We then write the following set of clauses:

$$u \oplus r \oplus d \oplus l = \begin{cases} false & \text{for all the black vertices} \\ true & \text{for the only white vertex} \end{cases}$$

- $\{\overline{u}, r, d, l\}$, $\{u, \overline{r}, d, l\}$, $\{u, r, \overline{d}, l\}$, $\{u, r, d, \overline{l}\}$, $\{u, \overline{r}, \overline{d}, \overline{l}\}$, $\{\overline{u}, r, \overline{d}, \overline{l}\}$, $\{\overline{u}, \overline{r}, d, \overline{l}\}$, $\{\overline{u}, \overline{r}, \overline{d}, l\}$ for a black vertex
- $\{u, r, d, l\}$, $\{\overline{u}, \overline{r}, d, l\}$, $\{\overline{u}, r, \overline{d}, l\}$, $\{\overline{u}, r, d, \overline{l}\}$, $\{u, \overline{r}, \overline{d}, l\}$, $\{u, \overline{r}, d, \overline{l}\}$, $\{u, r, \overline{d}, \overline{l}\}$, $\{\overline{u}, \overline{r}, \overline{d}, \overline{l}\}$ for the white vertex

Again, all the variables that do not make sense are replaced by "false".

There is another, chessboard-style formulation of the problem. Given a chessboard, tile it by dominoes, such that every square is covered by even number of tiles and one of the corners is covered by odd number of tiles. The formulation is illustrated on the right side of Fig. 2.

2 Preliminaries

Resolution We first give some definitions. A *literal* is either a propositional variable or the negation of propositional variable. A *clause* is a set of literals. It is satisfied by a truth assignment if at least one of its literals is true under this assignment. A set of clauses is *satisfiable* if there exists a truth assignment satisfying all the clauses.

Resolution is a proof system designed to *refute* given set of clauses, i.e., to prove that it is unsatisfiable. This is done by means of the resolution rule

$$\frac{C_1 \bigcup \{v\} \quad C_1 \bigcup \{\neg v\}}{C_1 \bigcup C_2},$$

i.e., we can derive a new clause from two clauses that contain a variable and its negation respectively. The goal is to derive the empty clause from the initial ones. For technical reasons only, we use the weakening rule

$$\frac{C}{C \bigcup \{v\}},$$

even though its use is not essential and can be avoided.

Anywhere we say we *prove* some proposition, we mean that first we take its negation in a clausal form and then use resolution to refute these clauses.

There is an obvious way to represent every resolution refutation as a directed acyclic graph whose nodes are labelled by clauses. The sources, i.e., the vertices with no incoming edges, are the initial clauses, and the only sink, i.e., the vertex with no outgoing edges, is the empty clause. If we reverse the directions of the edges, and consider the sink as a root and the sources as leaves we get a *branching program*. It is easy to see that it solves the following *search problem*, associated with the given set of unsatisfiable clauses: given an assignment, find a clause that falsifies it. Unfortunately, the reverse is not true, that is we cannot convert any branching program, solving the search problem, into a resolution proof.

As a matter of fact, there are polynomial-size branching programs, solving both problems from the paper. Of course, this does not contradict to our main result, as it shows that these branching programs cannot be transformed into resolution proofs.

In our proof we essentially use a representation of resolution proofs as *Prover-Adversary game*s, called further *Resolution Games*. This approach is introduced by Pudlak in [6]. A brief description follows.

Proofs as Games There are two players, named *Prover* and *Adversary*. An unsatisfiable set of clauses is given. Adversary claims wrongly that there is a satisfying assignment. *Prover*'s task is to convict him in lying. A *position* in the game is a partial assignment of the propositional variables. The game start from the empty position. Prover has two kind of moves:

1. She queries a variable, whose value is unknown in the current position. Adversary answers, and the position then is extended with the answer.

2. She forgets a value of a variable, which is known. The current position is then reduced, i.e., the variable value becomes unknown.

The game is over, when the current partial assignment falsifies one of the clauses. Prover then wins, having shown a contradiction.

As she can always win, simply querying consecutively all the variables and not forgetting anything, Adversary's task is to force Prover to use *big memory*, "big" meaning exponential in the number of variables. We assume that she keeps her strategy as a *list of ordered pairs* (position, move), where "position" and "move" have their natural meaning. Thus, it is enough for Adversary to use a strategy, which ensures big number of different possible positions, no matter how Prover plays.

The reduction from a resolution proof to Resolution Game should now be clear. Although trivial, we will not explain it here and refer to [6] for all the details. We should however note that a *deterministic* Adversary's strategy corresponds to a *single path* in the proof's graph. Therefore, he has to use a *randomised strategy* (called "super-strategy" in Pudlak's paper) in order to enforce a *big enough subgraph*.

It is very important to make the following conventions: Every time we say "Prover's strategy", we mean *wining strategy*, as only a wining strategy corresponds to a resolution proof. Every time we say "Prover ... in order to win" we also mean *wining strategy*, i.e., Prover is not interested in wining a single game, but any game, no matter how Adversary plays.

We can finally state the main results and explain informally the main ideas behind the proofs.

Main results and outline of the proofs. We prove the following two theorems:

Theorem 2.1 *Any resolution proof of the Mutilated Chessboard problem is of size* $2^{\Omega(n)}$.

A weaker version of the above theorem, with $\sqrt{n}$ in the exponent, is proven independently in [1].

Theorem 2.2 *Any resolution proof of Tseitin tautologies, based on* $n \times n$ *rectangular grid graph, is of size* $2^{\Omega(n)}$.

The general idea of the proofs is following:

We consider Resolution game. Clearly, Prover's queries are pairs of neighbouring squares, and Adversary's answers are dominoes, covering these pairs. A domino can be either "yes" or "no", with the natural meaning. We divide the chessboard into non-overlapping *constant-size* squares called *zones*. During the game every zone is either completely empty or completely covered by dominoes by Adversary. Here "completely" means the entire zone, except possibly few squares on the borders. In the first, *randomised*, phase of his strategy, Adversary first constructs many covers of the zone, depending on all the possible shapes of its neighbouring zones, and he then picks one of them *at random* and remembers it. These covers satisfy certain conditions that will be explained later in the paper, when proving the results. The second, *deterministic*, phase is the real game. When Prover queries a variable, i.e., a domino, inside an empty zone, Adversary puts the cover, already chosen in the first phase. He does not however reveal the cover to Prover, but only answer the question consistently with the cover. If Prover forgets *all the queried* variables inside a covered zone, Adversary removes the cover, so that the zone becomes empty again. Thus a zone is nonempty if and only if it contains at least one "significant" variable (the exact meaning of this is given in the detailed proof), whose value is kept by Prover. There are two main points in our proof:

1. Prover has to remember $\Omega(n)$ variable values at some point in the game in order to win. That is in any resolution proof we have a clause, containing linear in n number of variables. This can be proven on somewhat higher level, depending on the connection between zones, but not on their specific covers or particular shapes. A nice abstraction of that is *Tiling Games*. They are considered in a separate section.

2. Every two values, kept by Prover and belonging to different zones are *independent* of each other. Moreover, given any value, kept by Prover, there is a *constant probability*, bounded away from 0 and 1, that the value *agrees* with the first, randomised phase. These properties depend on randomised phase only, and on some specific properties of the zone covers, designed there. This can be thought as a *reduction* of Tilling Game to Resolution Game.

It is not hard to see that these two conditions imply an exponential lower bound on Provers's memory, and therefore on any resolution proof of the corresponding problem.

The rest of the paper is organised as follows. We first introduce *Tiling Games*. They allow us to work on the level of zones only, when proving the first main claim. We also prove an exponential lower bound for these games. After that, we show a reduction between Resolution Games and

Tiling Games that preserves the lower bound. This proves the second main point.

3 Tiling games

In this section, we introduce *tiling games* and prove some results about their complexity.

Definition of a general tiling game. The *board* of the game consists of $m \times m$ squares. Any of them is a perfect square, except the bottommost right one that has a dent on its right side. The board is shown on the left of Figure 3. The *tiles* of the game are squares. Their sides are of

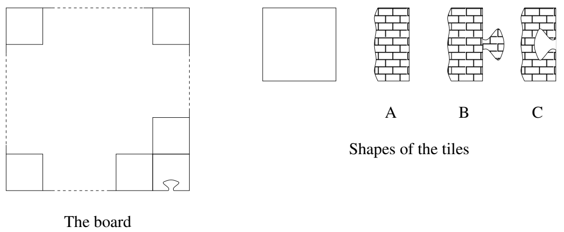

Figure 3. The tiling games

three kinds: "flat wall", "hump", and "dent", as shown on the right of Figure 3, pictures A, B, and C, respectively. Apart from its *shape*, every tile has also a *colour*, either red or blue. When we say *a general tiling game*, we mean a game where any set of shapes is allowed, whereas to get *a particular tiling game*, we fix this set. Thus, a (particular) tiling game is completely determined by its set of tiles. In both cases, every colour, either red or blue, is allowed for every shape.

In what follows, we will however consider only sets of tiles, having the property that they *cannot completely cover* the board. In particular, we put the restriction that the difference between the number of dents and the number of humps has to be even for any tile from the set. A trivial parity argument then implies the impossibility of tiling the board. We can also note that there are 41 such tiles, and therefore 2^{41} possible tiling games, as any subset of tiles defines a different one. We will however be interested in only two of them.

There are two players, whose names are *Prover* and *Adversary*. Adversary claims that there is a tiling of the board. Prover's task therefore is to force a *clear* contradiction, i.e., a tile on the board, which is *inconsistent* with one of its neighbours. In that case, she wins the game, which is played as follows: At the beginning the table is empty. At any round Prover starts by doing one of the following two:

1. She asks Adversary to put a tile on a particular square. He does so, and the round is over. We assume that Prover has infinite number of tiles of any kind (that is any allowed shape and any colour).
2. She removes any tile, already on the board. Adversary does not do anything, and the round is over.

The game is over, when Adversary is not able to play in the first case. That is, there is no tile, whose shape is consistent with the tiles, already on the board (note that the colour does not play any role here). Prover can always win by simply asking about all the squares and not removing anything. Adversary, knowing this, does not hope to play forever. His task instead is to force Prover to use *big memory*, no matter what she does.

Therefore, we need finally to explain how Prover "memorises" her strategy: The strategy is kept again as a *list of ordered pairs* $(\text{position}, \text{ move})$, where "position" and "move" have their natural meaning. It is now clear how Prover plays: In the beginning, she finds a pair, having its position-part empty. She then makes the move-part of the pair. A new position appears. Prover finds a pair, having the new position in its position part, and then makes the move-part, and so on... We need also the restriction, that every two pairs from Prover's list have to have different position-parts, that is Prover's strategy is *deterministic*.

We can now explain how Adversary enforces the use of *big memory.*

Adversary's strategy and general lower bounds. First of all, let us note, that Adversary's strategy *cannot be deterministic*, as Prover can query about all the squares in some fixed order, never removing anything from the board, thus wining the game in m^2 memory.

We will now describe a *randomised* strategy, which is *optimal* against any Prover's strategy.

The first, *randomised*, part is very simple. It involves *the colours only*. We choose the colour for *all the squares* independently, at random, with equal probabilities of $1/2$. During the game, when asked to put a tile on a particular square, Adversary always uses the initially chosen colour.

The second part is completely *deterministic*. It involves *the shapes of the tiles only*. To explain it, we need some definitions.

Definition 3.1 *Given a position, a* ***bad square*** *(for this position) is a square, such that the board, except this square, can be tiled. The* ***bad region*** *is the set of all the bad squares.*

In general, it is not even clear that a bad square exists for any position. From now on, we shall however consider only tiling games, which satisfy the following

Property 1 *The bad region for the starting position, i.e., an empty board, is* the entire board, *itself.*

Informally speaking, we would like to be able to move the "problematic" square from the south-eastern corner to any other position. We can make the following trivial observation:

Proposition 3.2 *Given a position where the bad region consists of two or more squares, Prover cannot win immediately, i.e., at this round.*

The next observation, although simple, is essentially the second part of Adversary's strategy.

Proposition 3.3 *Adversary can play in such a way, that the size of the bad component decreases by at most a constant factor after every round.*

Proof Let us denote the bad region by B. If Prover removes a tile from the board, the size of B remains the same or increases. Let us suppose now that Prover asks Adversary to put a tile on the empty square s. Adversary then tries all possible tiles $t_1, t_2 \dots t_k$, i.e., shapes consistent with the non-empty neighbours of s, as the colour has already been decided in the first part of the strategy. For each of these k possibilities, we denote the new bad region by $B_1, B_2 \dots B_k$. Let us now observe that any bad square for the initial position, b, has either to be s (if it is bad itself) or to belong to some B_j. The latter holds, because in tiling the entire board except b, there is a tile among the t_js, put on s, and then b certainly is in the corresponding B_j. Therefore, we have

$$|B_1| + |B_2| + \dots + |B_k| \geq |B| - 1.$$

It is now clear that Adversary has to take the most natural decision, that is to maximise the size of the new bad region. In this case

$$|B_{new}| \geq \frac{|B| - 1}{k} \geq \frac{|B|}{2k}.$$

This completes the proof, as k is less or equal to the number of all possible shapes of tiles, which is a constant (at most 41, as already mentioned). □

An important consequence is the following fact.

Proposition 3.4 *In any play of a tiling game there has to be a point, when* the bad region area *is* αm^2 *for some constant* α, strictly *between* 0 *and* 1. *At the same round* the border *of the bad area has to be of* length βm *for some positive constant* β *depending on* α.

Informally speaking, there must be a point, when the bad region is "big", i.e., quadratic in m. Naturally, in order to "surround" such a big area, we need a "big", i.e., linear in m, border. Of course, we need first to rigorously define the concepts mentioned in the statement, even though their meaning is intuitively clear.

Two squares on the board are *neighbours* if they have a common side. A *region* is an arbitrary set of squares. The *border* and the *complement* of the region R, $\partial(R)$ and $co(R)$, are defined as follows: $\partial(R)$ is the set of squares, having the property that each element in-there has a neighbour in R. $co(R)$ is all the rest, i.e., it contains every square that is in neither R nor $\partial(R)$. The *closure* of R is $\overline{R} = R \cup \partial(R)$. When we say "area" and "length", we really mean "number of squares".

We can now prove the proposition, itself.

Proof Let us observe that the area of the bad region goes from its initial value m^2 to 0 at the end, as it is a wining play for Prover (Proposition 3.2). Consider the first round, after which the area drops below $m^2/2$. After that round, it has to be bigger than $m^2/(2 \times 2 \times 41)$, according to Proposition 3.3. This proves the first part, with $\alpha \in [1/164, 1/2]$.

For the second part, we will use the following lemma whose proof is omitted, because of the space restriction.

Lemma 3.5 *For any region* R, $|\partial(R)|^2 \geq \min\left\{|\overline{R}|, \left|\overline{co(R)}\right|\right\}$.

Clearly, it implies the second claim in the proposition, with $\beta = \min\left\{\sqrt{\alpha}, \sqrt{1-\alpha}\right\}$. In our special case $\beta = \sqrt{\alpha}$, as $\alpha \leq 1/2$. □

Let us summarise what has been done so far: We have considered **a general tiling game**, under the only (rather weak) assumption that the bad region is the entire board at the starting position. We have proven that in any game played, there is a point, when the bad region has to have **a border linear in** m . We can note that we have not used the colours of the tiles in any way.

We can now formulate our second assumption.

Property 2 *In any position in the game, the number of tiles on the board is* linear *in the length of the border of the bad component.*

Adding this general, though still weak, assumption, to the first one, we can easily prove *an exponential* in m lower bound on Prover's memory.

Theorem 3.6 *In a tiling game, satisfying properties 1 and 2, any Prover's wining strategy is of size* $2^{\Omega(m)}$.

Proof According to Lemma 3.4, there is a point in the game, when the bad area is of size quadratic in m. At the same point, the border has to be of size$\Omega(m)$. By Property 2, the number of tiles on the board is $\Omega(m)$, too. The probability, that this position is consistent with the first part of Adversary's strategy, random colouring, is $1/2^{\Omega(m)}$. Therefore Prover has to have at least $2^{\Omega(m)}$ different position in the memory, as otherwise, there would be a choice of the colours, for which she does not win.□

At the end, let us note that our two lower bounds on the size of both a position in the game (linear) and Prover's memory (exponential) are tight. A simple divide-and-conquer algorithm yields *upper bounds* of $O(m)$ for the size of the position at any time and $2^{O(m)}$ for the memory Prover needs.

What remains to be done is to prove that our two assumptions are indeed correct for the concrete tiling games we are interested in.

Length lower bounds for particular games We shall first define the two games.

1. **Tseitin** is the tiling game, having as a set of tile-shapes all the shapes for which the *difference* between the number of dents and the number of humps is *even*.

2. **Mutilated Chessboard** is the tiling game, having as a set of tile-shapes all the shapes for which the number of dents *equals* the number of humps.

Clearly, Mutilated Chessboard game is a restricted version of Tseitin game. Thus, every lower bound for the former game applies to the latter, too. On the other hand, one could expect that proving lower bounds for Mutilated Chessboard game would be much harder. It is indeed the case, as the reader will see. This is the reason, we spend most of the rest of the paper on Mutilated Chessboard game rather than on Tseitin one.

First of all, let us observe that both games trivially fulfill the first assumption, saying that initially, the bad region is the entire board.

Thus only the second assumption, namely that at any round, the number of tiles on the board is linear in the border-length of the bad region, is to be checked.

We start with the easier, Tseitin, tiling game. In this case, the deterministic part of Adversary's strategy can be simplified. The key observation is that we can always keep the bad region *isolated.*

Definition 3.7 *The bad region is* ***isolated*** *iff it is* separated *by tiles from any other region of the board, consisting of empty squares. In other words, a neighbour of a bad square is either another bad square or a tile, but never a square, which is not bad.*

In general, we need to consider all the connected components of empty squares. We can keep the following invariant: exactly one of them is bad and the others are *good*, i.e., they can be tiled. It can be easily proven that the only component having one more dent that humps is the bad one, and moreover any component, having equal number of dents and humps, is good. When Prover asks a question, she may disconnect the component, where the question is, into (at most four) other components. Conversely, if Prover removes a tile from the board, she may join some previously disconnected components into a new one. Adversary needs to be careful only in the case when the Prover's question disconnects the current bad component. If so, Adversary answers in such a way that *the biggest* of the new obtained components becomes *bad* and the rest become good. It is easy to see that Adversary can always do that by a tile, consistent with the neighbours of the queried square. After any round of the game, the bad component can decrease by a factor of four at most, thus Proposition 3.3 holds, and so does Proposition 3.4 with $\alpha \in [1/16, 1/2]$ and $\beta = 1/4$. As *the bad region* is always *isolated*, its border consists of tiled squares only. Therefore the second assumption is fulfilled, and Theorem 3.6 then applies, giving us the following:

Lemma 3.8 *Prover needs at least* $2^{\frac{m}{4}}$ *memory cells in order to win Tseitin tiling game.*

Let us now consider the other tiling game, Mutilated chessboard one. It is now not so easy as before, as *the bad region* does *not* need to be *isolated*. This is illustrated on

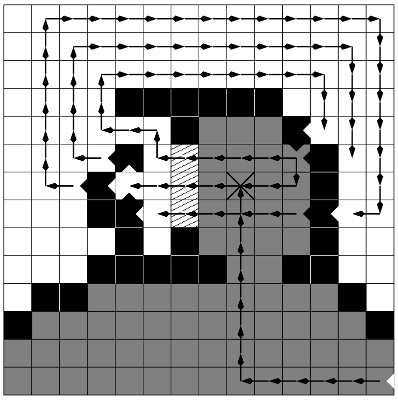

Figure 4. Tiling to flow correspondence

Figure 4. Black squares are the tiled ones. We call them also *marked* squares. The bad region is shown in gray. Its border contains not only marked squares, but also some empty squares, which are shown dashed. One could, in general, think that it is possible, in some clever way, to "surround" a "big" bad area, using only "few" tiles. Our intuition however tells us, that it should be impossible. If the border of the bad region is coarse, it would be possible to "push" a problematic square through it, thus "extending" the bad region, which is impossible by its definition. The following argument formalises the intuition.

We shall first explain the connection between of a position in the Mutilated Chessboard game to max–flow in a graph. The vertices of the graph are the empty squares and there is an edge between any two neighbours. We call *a* *source* an empty square such that there are more dents inside it than the dents in the neighbours, adjacent to the considered square, i.e., the humps of the square if it were tiled. The difference between the number of these two numbers is the *capacity* of the source. On the picture, there are four sources of capacity one and one source of capacity two. They are the squares with outgoing arrows only. In general, when counting them, we take into account the capacity, so that we can say that there are six sources on the picture. If we exchange "dents" by "humps" and vice versa in the above definition, we get the definition of a *sink*. There are five sinks on the picture that are exactly the squares with incoming edges only. Obviously, the number of sources is always greater by exactly one than the number of sinks during the game. It is clear that the bad squares and only they have the following property: if we choose one of them as a sink, so that the number of sources equals the number of sinks, there is a max–flow of capacity equal to the number of sources. An example is given on Figure 4, where the crossed square is chosen and a max–flow (of value 6) is shown by the arrows. It is also straightforward to convert the flow into the corresponding tiling an vice versa.

Proposition 3.9 *In any position of Mutilated Chessboard tiling game, the number of empty border squares is a constant fraction of the total number of border squares.*

Proof We now consider the border of the bad region. What we need to prove first is that the empty squares, belonging to it, are not "too many", namely they are less than the number of sources.

The proof uses a max–flow - min–cut argument. Let us introduce a new, artificial vertex A, and a directed edge from every empty border square to A. Let us put the new vertex as a sink of capacity one, and denote the number of sources by k. We now claim that there is no flow of value k in the new graph. Suppose there were. Then it had to go trough one of the new edges. But then we could "stop" it in the corresponding border squares that would imply this square is bad - a contradiction.

Since the max–flow equals the min–cut, there has to be a cut of size less than k. Let us take one such cut, and call the *sources side "right"* and the *sinks side "left*".

We first need to show that the artificial vertex A is not contained in the cut[1] (note that a cut can, in general, contain not only edges but also vertices, having capacities). Suppose that the cut contained A. Consider the part of the cut when restricted to the original graph, i.e., before adding A and the edges from any empty border square to A. This part of the cut must be of capacity at least $k-1$ as it separates the k sources from $k-1$ sinks, and there is a max–flow of value $k-1$ in the original graph. Therefore any cut containing A is of capacity at least k. This implies that it cannot be minimal as there is no flow of value k in the new, containing A graph.

We can now see that all the bad squares are on the right side. Suppose there were at least one on the other side. But this implies that the size of the cut is greater or equal to k, because there is a flow of size k if a bad square is taken as a sink - a contradiction.

Let us denote the sets of border squares on the left/right side by L/R respectively. What we have proven so far is shown on Fig. 5. We should however note that there are

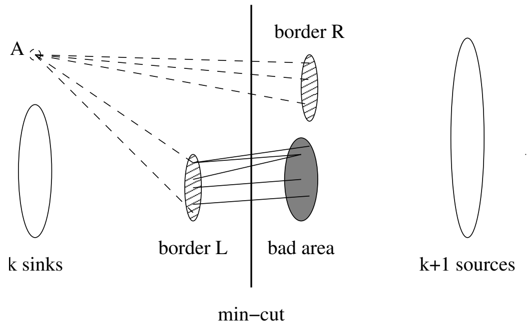

Figure 5. Max–flow - min–cut argument

two properties, which we "ignore" in our proof, because we do not need them. First, as a matter of fact, it can be proven that all the border squares are on the left side of the cut. Therefore, $R = \emptyset$, and the cut contains no artificial edges. Second, it can be shown that the bad area is connected.

The cut we consider contains at least the following edges: exactly one edge from every vertex in R to the new vertex and at least one edge from every vertex in L to some bad vertex (as every vertex in L is on the border of the bad area). This implies that $|L| + |R| < k$, that is the number of border squares is less than the number of sources.

We can finally prove that the second property, saying that at any round, the number of tiles on the board is linear in the border-length of the bad region, is fulfilled. Given a position on the board, denote the border-length of the bad region by l. Suppose the *number of sources* is *at most* $l/2$. There are then at most that many empty squares on the border, thus the number of marked squares, that is the number of tiles on the board, is at least $l/2$. Suppose now the opposite, the *number of sources* is at *least* $l/2+1$. Observe now that every marked square generates at most two sources, as any tile has at most two humps. Thus, there have to be at least $l/4$ tiles on the board. This completes the argument.□

As proven before, the above proposition implies

Lemma 3.10 *Prover needs $2^{\Omega(m)}$ memory cells in order to win Mutilated Chessboard tiling game.*

[1] We thank Mikhail Alekhnovitch for pointing out that we have forgotten to include this part of the argument in an earlier version of the paper.

4 Reduction

In this section, we show how to reduce *Resolution game,* played on a c*hessboard,* into *Tiling game*. To understand what “reduction” really means in this context, we need to look at Fig. 6.

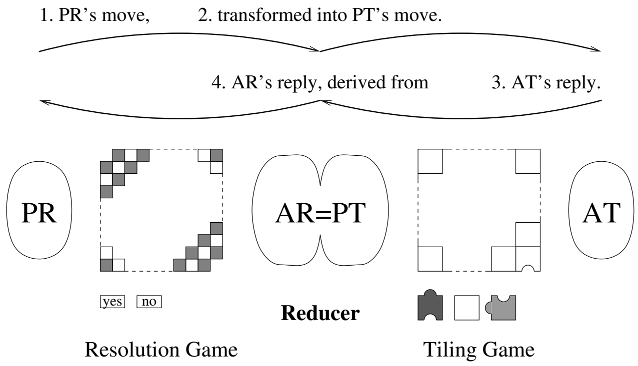

Figure 6. The general schema of the reduction

The resolution game is played by Prover Resolution (PR) and Adversary Resolution (AR), while the tiling game is played by Prover Tiling (PT) and Adversary Tiling (AT). We think of AR and PT as a single person, named *Reducer*, who carries the *reduction*. As shown on the figure, he first looks at the PR’s move in the resolution game. He then transforms it into PT’s move in the resolution game and plays it there. After having got AT’s reply, Reducer transforms it into AR’s move and replies by it to the initial PR’s move in the resolution game.

Thus, one can think that the real game is played between PR and AT. We already have a particular AT’s strategy which forces an exponential lower bound on any PT’s strategy. We will prove, that this important property can be carried trough the reduction, that is to imply an exponential lower bound on any PR’s strategy, too. We will only consider the reduction of Mutilated Chessboard problem, which is technically harder. That is why, we describe it in full detail, leaving the reduction for Tseitin tautologies to the reader.

Mutilated chessboard As we mentioned before, we first divide the chessboard into non-overlapping *constant-size squares*, called further *zones*. In our rigorous proof, we use 48×48 squares. A *zone* in Mutilated Chessboard problem corresponds to a *square* in Mutilated Chessboard tiling game. We also “move” one of the missing squares near to the other as shown on Fig. 7 for a $(48n + 2) \times (48n + 2)$ chessboard. We first define what a *zone* is. It is a “big”, 48×48, square, with “few” small squares cut off. We need to explain how the missing squares can exactly appear.

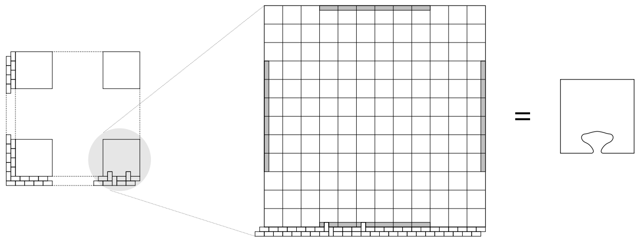

Figure 7. The reduction: zones, corresponding to tiles

We divide a zone into 12×12 smaller, 4×4, squares, further called *sub-zones*, as done in the middle of Figure 7. *Missing squares* can only appear on the *sides* of a zone, inside the four dashed “bands” which are the border part of the four gray five-sub-zone areas. Moreover, there are only the following possible shapes:

1. No missing squares (Fig. 8A). This corresponds to a *flat wall*, in the tiling game.

2. Two neighbours, of *different colour*, belonging to a sub-zone, and *not* being the two *middle* squares (Fig. 8B). In the tiling game, this corresponds to a *flat wall*, similar to the previous case.

3. Two squares of *the same colour,* belonging to sub-zones, that are a sub-zone away from each other (Fig. 8C). *Two black* missing squares correspond to a *hump* in the tiling game*,* whereas *two white* missing squares correspond to a *dent*.

4. Four missing squares, that are a *combination* of the previous two cases, and, moreover, no two mutilated sub-zones can be neighbours (Fig. 8D). Again, if *two more white* squares than black ones are missing, this is *a dent* in the tiling game. The symmetric case, i.e., *two more blacks*, corresponds to *a hump*.

Figure 8 shows all possible shapes of a *zone border*. They imply all possible *connections* between two *neighbouring zones*. In particular, two neighbours can have only 0, 2 or 4 dominoes in common, and, moreover, these can only appear as explained above.

We are now almost ready to explain the essence of the section, namely Reducer’s algorithm. The last, but the most important concepts, we need, are the different kind of questions we have in Resolution Game. At any round in the game, we have a partial tiling of the chessboard. The tiling satisfies the condition that any zone is either *completely empty* or *completely covered* by dominoes from Adversary’s point of view. There are the following three kind of Prover’s questions:

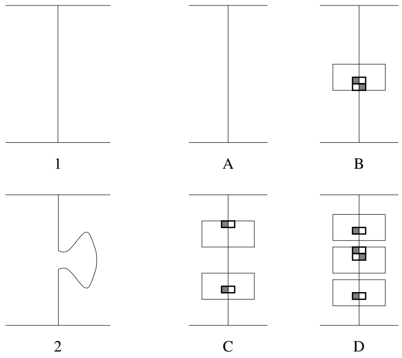

Figure 8. The reduction: possible shapes of zones and connections between them

Definition 4.1 ***Dummy*** *question is a question about a domino, connecting two neighbouring zones, and within* 3 *sub-zones (that is* 12 *squares) from one of the two common corners of the zones.*

The answer to an impossible question is always "no", so we can assume Prover gets them for free, and she never asks such a question.

Definition 4.2 ***Forced*** *question is a question, which is not dummy, but the domino involved affects the current partial tiling.*

The answer clearly depends on the current tiling.

Definition 4.3 ***Open*** *question is a question, which is neither dummy nor forced.*

That is, an open question is about a domino:

1. The domino does not intersect the current (partial) cover.
2. It either is completely inside an *empty zone* or connects *two empty neighbouring zones*. In the latter case, the domino is also 12 squares away from any *zone corner.*

We should note that "dummy" is a static concept, i.e., it does not depend on the current partial tiling, whereas the concepts "forced" and "open"are dynamic.

We should also note that a forced/open question may be assigned to any of two neighbouring zones. Sometimes, we will need to assign a question to a particular zone. We will then use the following deterministic rules:

1. If the question is open, we associate it to either the *right* zone (if the border is vertical) or the *bottom* one (if the border is horizontal).
2. If the question is forced, we associate it to the zone, which has been covered *first*, starting from the last point in the game, when both zones were empty. In this way we ensure that the question was open to its zone at that time.

We can now describe

Reducer's algorithm

The randomised phase. For any zone, further called "current", we do the following: For any possible shape of any combination of its *nonempty* neighbours and any possible connection to its *empty* neighbours, either dent, hump or flat wall, we design a set of tilings of the current zone. These tilings have to have the property that not all of them agree on any open question associated to the current zone. It is very important to note, that the number of all the tilings is a *constant*. Adversary then chooses one of the tilings uniformly at random and remembers the choice through the entire Resolution Game. What remains to be proven is that a set of tilings, having the desired properties exists. The proof is a long case analysis, omitted because of the space restriction.

The deterministic phase. We should first note that at any round in Resolution Game, there are some "yes" and/or "no" dominoes on the mutilated chessboard. These are visible to both Prover Resolution and Reducer, who is also Adversary Resolution. Apart from them, there is a partial tiling of the board, which is visible only to Reducer. This tiling is consistent with all the "yes"/"no" dominoes until the end of the game, when Reducer gives up. In Tiling Game, there are some tiles on the board. Moreover, there is a correspondence between any tile and the corresponding zone in Resolution Game, as explained at the beginning of the section.

Let us now consider the four stages of the reduction for the two possible Prover Resolution's moves, either asking a question or forgetting:

1. Asking a question about a domino.
   (a) Prover Resolution asks a question in Resolution Game.
      i. The question is forced. Reducer answers immediately, without going to the next stage.
      ii. The question is open, and therefore it is the first one such question about the considered empty zone. Reducer switches from Adversary Resolution to Prover Tiling, thus going to the stage (b)
   (b) Prover Tiling asks a question in Tiling Game.

(c) Adversary Tiling puts a tile on the board.

(d) Given that tile and the neighbouring zones, Reducer gets the already chosen (in the randomised phase) cover of the zone and puts it on the mutilated chessboard, not revealing it to Resolution Prover. Reducer then switches back to Adversary Resolution and answers to Prover Resolution's question consistently with the cover just put.

2. Forgetting a domino, which is already on the mutilated chessboard.

   (a) Prover Resolution forgets a domino, already on the chessboard.

      i. The domino is not the only “yes”/”no” domino inside the corresponding zone. Reducer does not do anything else.

      ii. The domino is the only “yes”/”no” domino inside that zone. Reducer goes to stage (b).

   (b) Reducer, acting as Prover Tiling, removes the corresponding tile in the tiling game. After that, acting as Adversary Resolution, he forgets the cover of the zone, so that it becomes empty again.

The important lemma, that follows from our construction is the following:

Lemma 4.4 *At any round of Resolution Game, we can take a set of “yes”/”no” tiles, no two of them belonging to the same zone. They are independent and the probability that each of them agrees with the randomised phase is a constant, bounded away from both* 0 *and* 1.

Proof Because of the way we associate a question to a zone at the time when the zone was covered, the question was open for it. We can now use the fact, that the cover was chosen at random, among the set of covers, such that not all of them agree on any open question.□

That is enough to ensure that Theorem 3.6 applies, with the two colours, corresponding to the two possible answers to the chosen questions. The probabilities now are not $1/2$ and $1/2$, but some (small) constants, different from 0 and 1. This however does not affect the argument, so that the exponential lower bound is carried from Tiling Game through the reduction to Resolution Game.

Acknowledgements We would like to thank Mikhail Alekhnovich for finding the missing point already mentioned.

References

[1] M. Alekhnovich. Mutilated chessboard is exponentially hard for resolution. Manuscript, 2000.

[2] E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. In *Symposiom on Theory Of Computing*, pages 517–526, 1999.

[3] S. Buss and G. Turán. Resolution proofs of generalized pigeonhole principles. *Theoretical Computer Science*, 62:311–317, 1988.

[4] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[5] J. McCarthy. A tough nut for proof procdures. Stanford Artifical Intelligence Project Memo 16, July 1964.

[6] P. Pudlák. Proofs as games. *American Mathematical Monthly*, June–July 2000.

[7] G. Tseitin. On the complexity of derivation in the propositional calculus. *Studies in Constructive Mathematics and Mathematical Logic, Part II*, 1968.

[8] A. Urquhart. Resolution proofs of matching principles. Manuscript, 1998.